

Краткая пояснительная записка к проекту

“Компактная умная ферма с WiFi-управлением”

1 Состав проектной команды и вклад участников

- **Лепин Владислав Дмитриевич** (Б01-304) — программирование ESP32, разработка архитектуры программной части, пайка, проектирование и сборка схемы, подготовка презентации, дизайн корпуса.
- **Шарипов Акрам Рустемович** (Б01-304) — разработка Android-приложения на Java, проектирование всех 2D и 3D моделей, пайка электронных компонентов.
- **Николаев Николай Александрович** (Б01-306) — разработка серверной части для коммуникации между фермой и приложением, сборка конструкции фермы.

2 Причины выбора проекта

Основная мотивация — создание практически полезного устройства для повседневного использования, а не просто демонстрационного прототипа. Автоматизированная ферма решает практическую задачу обеспечения растений оптимальными условиями с минимальным участием человека. Проект позволил применить знания из программирования, электроники и 3D-моделирования.

3 Цель и задачи проекта

Цель: создание компактной автоматизированной умной фермы для выращивания сельскохозяйственных культур с возможностью удаленного мониторинга и управления через мобильное приложение. Система должна обеспечивать автоматический полив, контроль микроклимата, освещения и передачу данных в режиме реального времени.

Задачи:

- Разработать модульную систему автоматизации полива, контроля микроклимата и освещения
- Создать корпус фермы с креплениями для компонентов
- Спроектировать электронную схему на базе ESP32
- Разработать программное обеспечение с алгоритмами управления
- Создать мобильное приложение для удаленного управления
- Реализовать серверную часть для коммуникации

4 Описание продукта

IoP-Farm (Internet of Plants Farm) — автоматизированная мини-ферма на базе ESP32 с функциями:

- **Автоматический полив:** включение насоса R385 по расписанию
- **Контроль температуры:** датчик DHT22 отслеживает температуру, система включает тепловую лампу при необходимости
- **Управление освещением:** фитолампа работает по заданному расписанию
- **Мониторинг уровня воды:** датчик HC-SR04 контролирует уровень воды в резервуаре
- **Мобильное приложение:** мониторинг показателей и управление настройками

Компоненты:

- **Микроконтроллер:** ESP32 WROOM 32
- **Датчики:** DHT22 (температура и влажность воздуха), DS18B20 (температура воды), HC-SR04, FC-28 (влажность почвы), KY-018 (освещенность), YFS401 (расход воды)
- **Исполнительные устройства:** насос R385, тепловая лампа, фитолампа (LED)
- **Блок питания:** 12V адаптер питания
- **Корпус:** акриловое стекло 6 мм и 3D-печатные детали

5 Процесс решения поставленных задач

5.1 Программная часть

Программное обеспечение построено на принципах ООП и модульной архитектуры:

- **Датчики:** абстракция через интерфейс `ISensor`
- **Актуаторы:** все устройства реализованы через интерфейс `IActuator`
- **Стратегии:** определяют логику работы на основе данных от датчиков
- **Менеджеры:** `WiFiManager`, `MQTTManager`, `ConfigManager`, `OTAManager`, `WebServerManager`, `SensorsManager`, `ActuatorsManager`
- **Мобильное приложение:** Java-приложение для Android, взаимодействующее через сервер по TCP. Сервер взаимодействует с фермой по MQTT

Все менеджеры, взаимодействуя между собой, управляют работой фермы.

Исходный код проекта доступен в репозитории: <https://github.com/Skifv/IoP-Farm>

5.2 Аппаратная часть

Плоские детали корпуса изготовлены лазерной резкой акрилового стекла, сложные формы — 3D-печатью. Соединение деталей выполнено дихлорэтаном с фиксацией струпцинами. Электронные компоненты соединены по разработанной схеме и подсоединены к ESP32.

Все спроектированные модели доступны в репозитории: <https://github.com/Skifv/IoP-Farm>

6 Анализ существующих аналогов

Найденные нами аналоги имели схожий функционал, но небольшие различия: наличие/отсутствие полива/обогрева/освещения по расписанию/команде. Также в некоторых отсутствует контроль уровня воды или расхода воды. В одном из аналогов есть камера и визуальный контроль растений.

Отличительные особенности нашего проекта:

- Комплексная автоматизация: полив, обогрев и освещение в одной системе. Ни один из найденных нами аналогов не включал такого разнообразия компонентов
- Контроль расхода воды для точного полива заданного объема. В остальных проектах полив осуществляется на заданный интервал времени, что, безусловно, имеет большие минусы
- Модульная архитектура, позволяющая легко расширять функциональность
- Возможность OTA-обновления программного обеспечения

Здесь можно найти проанализированные аналоги:

1. <https://github.com/sunny-zuo/smart-garden>
2. <https://github.com/fruxefarms/FruxePi>
3. <https://github.com/PatrickHallek/automated-irrigation-system>

7 Процесс проектирования и изготовления

Процесс создания IoP-Farm включает в себя 3 этапа (то, что программирование находится на третьем месте, не определяет его порядок):

1. Проектирование:

- Разработка архитектуры системы, определение компонентов
- Проектирование электрической схемы подключения
- Моделирование деталей корпуса в FreeCAD и 3D-моделей в Компас 3D

2. Изготовление:

- Лазерная резка акрилового стекла, 3D-печать креплений
- Сборка корпуса с использованием дихлорэтана
- Монтаж электронных компонентов и пайка

3. Программирование:

- Разработка ПО для ESP32 на C++
- Создание Android-приложения на Java
- Разработка сервера на C++
- Настройка MQTT-брокера

8 Процесс тестирования и его результаты

Тестирование проводилось при помощи разработанной системы дистанционного логирования процессов, происходящих на микроконтроллере.

Использовались команды с мобильного приложения и анализировались логи от фермы через сервер. Результаты показали стабильную работу системы:

- Полив активируется при помощи команд, расписание работает корректно
- Система освещения и обогрева работают согласно выстроенным настройкам
- Показания датчиков корректно отображаются в приложении