

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка автоматизированной системы полива (2)»
по дисциплине «Проектный практикум»

Заказчик: Фамилия И.О.

Папуловская Наталья

Владимировна

Куратор: Фамилия И.О.

Изотов Илья

ученая степень, ученое звание, должность

Николаевич

Студенты команды __ GAZON_

Колотов Виталий

Фамилия И.О.

Владимирович

Фамилия И.О.

Вайс Владислав

Валерьевич

Екатеринбург, 2025

СОДЕРЖАНИЕ

| | |
|--|-----------|
| ВВЕДЕНИЕ..... | 3 |
| 1 Основная часть | 6 |
| 1.1 Роль и вклад каждого участника проекта..... | 6 |
| 1.2 Анализ требований заказчика и пользователей. Backlog разработки.... | 7 |
| 1.3 Анализ аналогов разрабатываемого продукта..... | 8 |
| 1.4 Архитектура программного продукта..... | 8 |
| 1.5 Методология разработки и тестирование..... | 9 |
| ЗАКЛЮЧЕНИЕ | 13 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 16 |
| ПРИЛОЖЕНИЕ А (обязательное) Расчет энергопотребления системы..... | 17 |

ВВЕДЕНИЕ

Целью данного проекта является разработка автоматизированной системы полива растений на основе микроконтроллера ESP, датчика влажности почвы и управляемой помпы. Система предназначена для поддержания оптимального уровня влажности грунта без необходимости постоянного контроля со стороны пользователя.

Основные задачи проекта включают:

- а) анализ современных решений в области автоматизированного полива и выбор оптимальных компонентов,
- б) разработку аппаратной части системы, включающей датчик влажности почвы, управляющую помпу и микроконтроллер ESP,
- в) создание программного обеспечения для обработки данных с датчиков, управления поливом и интеграции с возможностью удаленного мониторинга,
- г) тестирование системы в различных условиях эксплуатации для проверки её надежности и эффективности,
- д) доработка системы с возможностью подключения дополнительных модулей, таких как осветительные лампы для регулировки уровня освещенности растений.

Автоматизированные системы полива приобретают все большую популярность в условиях роста интереса к умным технологиям в сельском хозяйстве, садоводстве и домашнем растениеводстве. Ручной полив требует постоянного внимания и может быть неэффективным из-за неравномерного распределения влаги или человеческого фактора [3]. Разработка доступной и надежной системы автоматического полива на базе микроконтроллера ESP позволяет решить следующие проблемы:

- а) оптимизация расхода воды – система подает влагу только при необходимости, что особенно важно в условиях ограниченных ресурсов,
- б) повышение урожайности – поддержание оптимальной влажности способствует здоровому росту растений,
- в) удобство эксплуатации – система работает автономно, снижая трудозатраты пользователя,
- г) масштабируемость – возможность интеграции дополнительных датчиков и модулей (например, освещения) делает систему гибкой и адаптируемой под разные задачи.

Разрабатываемая система может быть использована в следующих сферах:

- а) домашнее растениеводство – для автоматического ухода за комнатными растениями, рассадой, мини-огородами,
- б) сельское хозяйство – в теплицах, на небольших фермах и приусадебных участках,
- в) ландшафтный дизайн – для поддержания влажности газонов, клумб и других зеленых насаждений,
- г) образовательные и исследовательские проекты – в качестве учебного стенда для изучения IoT-технологий и автоматизированных систем управления.

По завершении проекта ожидается создание работоспособного прототипа автоматизированной системы полива, обладающей следующими характеристиками:

- 1) автономная работа на основе показаний датчика влажности,
- 2) возможность ручной и автоматической настройки параметров полива,
- 3) интеграция с мобильным приложением или веб-интерфейсом для удаленного управления,
- 4) энергоэффективность и низкая стоимость компонентов, обеспечивающие доступность системы для широкого круга пользователей.

В перспективе система может быть дополнена модулем освещения, датчиками температуры и углекислого газа, что расширит её функционал и повысит эффективность контроля за микроклиматом. Реализация данного проекта позволит создать экономичное и практическое решение для автоматизации полива, соответствующее современным тенденциям в области "умного" сельского хозяйства и IoT-технологий.

1 Основная часть

1.1. Роль и вклад каждого участника проекта

В разработке автоматизированной системы полива участвуют два человека, каждый из которых выполняет определенные задачи для достижения общей цели.

Участник 1:

- а) отвечает за разработку программной логики микроконтроллера ESP (написание кода для считывания данных с датчика влажности, управления помпой и обработки сигналов),
- б) разрабатывает алгоритм автоматического полива на основе пороговых значений влажности,
- в) обеспечивает интеграцию с Wi-Fi для удаленного мониторинга (например, через MQTT или REST API),
- г) тестирует работу датчиков и исполнительных устройств, выявляет и исправляет ошибки в коде.

Участник 2:

- а) занимается разработкой пользовательского интерфейса, телеграмм-ботом для управления системой,
- б) настраивает серверную часть (если требуется облачное хранение данных или удаленное управление),
- в) разрабатывает систему уведомлений (например, оповещения о критически низком уровне влажности),
- г) проводит нагружочное тестирование и проверяет отказоустойчивость системы.

Оба участника совместно:

- 1) проводят анализ требований и составляют backlog,
- 2) участвуют в выборе компонентов и архитектурных решений,
- 3) тестируют систему на различных этапах разработки,

4) документируют процесс и фиксируют изменения в коде (используя Git).

1.2. Анализ требований заказчика и пользователей. Backlog разработки

Основные требования заказчика:

- а) система должна автоматически поливать растения при снижении влажности ниже заданного порога,
- б) возможность ручной настройки параметров (интервалы полива, длительность подачи воды),
- в) удаленный мониторинг через Wi-Fi (веб-интерфейс или мобильное приложение),
- г) энергоэффективность и низкая стоимость компонентов (Приложение А).

Пользовательские требования:

- а) простота настройки и использования,
- б) надежность работы без ложных срабатываний,
- в) возможность расширения функционала (например, добавление датчиков температуры и освещенности).

Backlog проекта:

- 1) анализ и выбор компонентов [1],
- 2) подбор датчика влажности [5],
- 3) выбор помпы (12V водяная помпа с управлением через реле),
- 4) определение микроконтроллера (ESP8266 или ESP32),
- 5) разработка прототипа,
- 6) сборка схемы подключения датчика и помпы,
- 7) написание базового кода для считывания данных и управления поливом [2,4],

```
def main() -> None:
    """Запуск бота."""
    bot = PlantMonitorBot()

    # Создаем Application и передаем ему токен бота
    application = Application.builder().token(BOT_TOKEN).build()

    # Регистрируем обработчики команд
    application.add_handler(CommandHandler("start", bot.start))
    application.add_handler(CommandHandler("help",
bot.help_command))
    application.add_handler(CommandHandler("auth", bot.auth))
    application.add_handler(CommandHandler("status",
bot.get_system_status))
    application.add_handler(CommandHandler("moisture",
bot.get_moisture))
    application.add_handler(CommandHandler("light",
bot.get_light))
    application.add_handler(CommandHandler("pump_on",
bot.pump_on))
    application.add_handler(CommandHandler("pump_off",
bot.pump_off))
    application.add_handler(CommandHandler("pump_time",
bot.pump_time))
    application.add_handler(CommandHandler("set_schedule",
bot.set_schedule))
    application.add_handler(CommandHandler("get_schedule",
bot.get_schedule))
    application.add_handler(CommandHandler("cancel_schedule",
bot.cancel_schedule))
    application.add_handler(CommandHandler("panel",
bot.show_control_panel))

    # Регистрируем обработчик нажатий на кнопки
```

```
application.add_handler(CallbackQueryHandler(bot.button_handler))

# Запускаем бота
application.run_polling()

if __name__ == "__main__":
    main()
```

- 8) программная реализация,
- 9) разработка алгоритма автоматического полива,
- 10) настройка Wi-Fi и удаленного доступа,
- 11) тестирование и отладка,
- 12) проверка корректности работы датчика,
- 13) тестирование отклика системы в разных условиях,
- 14) разработка интерфейса,
- 15) создание телеграмм-бота для управления,
- 16) доработка и масштабирование,
- 17) добавление модуля освещения (если требуется),
- 18) оптимизация энергопотребления.

1.3. Анализ аналогов разрабатываемого продукта

На рынке существуют различные системы автоматического полива, как коммерческие, так и DIY-решения:

Готовые коммерческие системы (например, Gardena, Xiaomi)

Плюсы: высокая надежность, удобный интерфейс.

Минусы: высокая стоимость, ограниченная гибкость настройки.

Arduino- и ESP-проекты (открытые решения на GitHub)

Плюсы: низкая стоимость, возможность кастомизации.

Минусы: требуют навыков программирования и сборки.

Вывод: Разрабатываемая система сочетает преимущества DIY-решений (гибкость, низкая стоимость) с удобством удаленного управления, что делает её конкурентоспособной.

1.4. Архитектура программного продукта

Система состоит из следующих компонентов на аппаратном уровне можно увидеть на рисунке 1:

- 1) микроконтроллер ESP (ESP8266/ESP32),
- 2) датчик влажности почвы,
- 3) водяная помпа с реле,
- 4) блок питания (12V для помпы, 5V для ESP).



Рисунок 1 – Компоненты системы

Программный уровень:

- 1) прошивка для ESP (написанная на C++),
- 2) алгоритм автоматического полива (пороговое управление),
- 3) Wi-Fi-модуль для передачи данных.

Пользовательский интерфейс:

- 1) телеграмм-бот на рисунке 2,

2) REST API для взаимодействия с микроконтроллером.

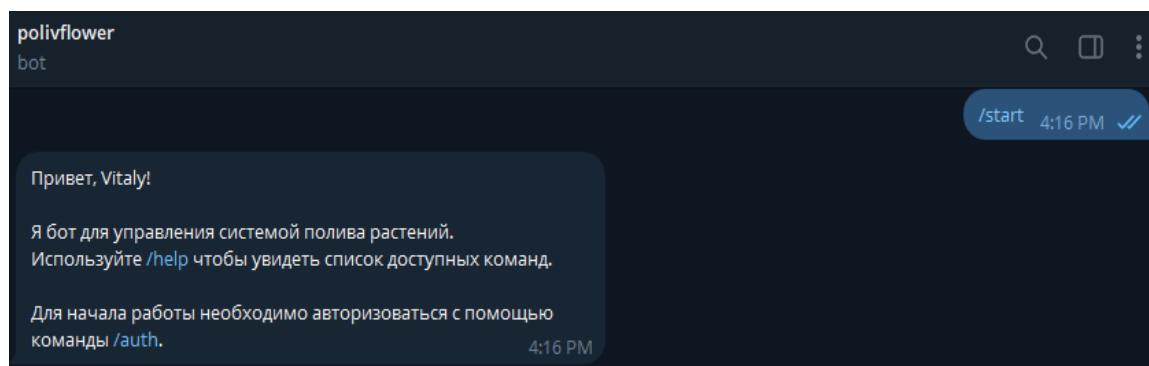


Рисунок 2 – Бот

Обоснование архитектуры:

Использование ESP позволяет реализовать Wi-Fi без дополнительных модулей. Релейное управление помпой обеспечивает простоту и надежность. Веб-интерфейс делает систему доступной с любого устройства. Схема подключения показана на рисунке 3.

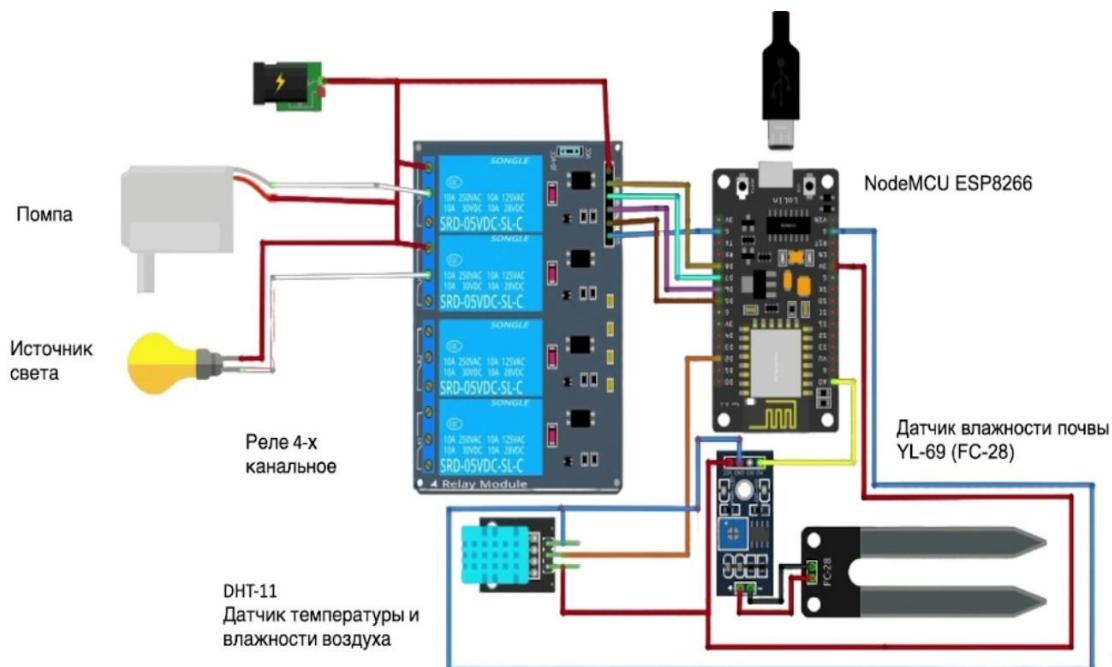


Рисунок 3 – Схема подключения

1.5. Методология разработки и тестирование

Методология:

Применяется гибкая методология (Agile/Scrum) с итеративными доработками.

Процесс тестирования:

- а) проверка работы датчика влажности (калибровка, точность измерений),
- б) тестирование реле и помпы (включение/выключение по сигналу).

Интеграционное тестирование:

- а) проверка связи ESP с Wi-Fi,
- б) тестирование алгоритма полива (корректность срабатывания).

Системное тестирование:

- а) работа в реальных условиях (разные типы почв, температура),
- б) проверка отказоустойчивости (например, при обрыве датчика).

Выявленные ошибки и их исправление:

Ложные срабатывания датчика - добавление фильтрации данных (усреднение значений). Перегрев помпы при длительной работе - ограничение времени непрерывной работы. Проблемы с Wi-Fi соединением - оптимизация кода и настройка переподключения.

ЗАКЛЮЧЕНИЕ

Разработанная автоматизированная система полива успешно соответствует ключевым требованиям, поставленным заказчиком и конечными пользователями.

Основная цель проекта – создание автономной системы, способной поддерживать оптимальный уровень влажности почвы без постоянного вмешательства человека – достигнута в полной мере.

Система обеспечивает:

- а) автоматический полив на основе данных с датчика влажности, что исключает необходимость ручного контроля,
- б) гибкость настройки, позволяя пользователю задавать пороговые значения влажности и длительность полива в зависимости от типа растений,
- в) удаленный мониторинг через Wi-Fi, что делает управление удобным и доступным с любого устройства,
- г) энергоэффективность благодаря использованию низковольтных компонентов и оптимизированному алгоритму работы.

Однако в ходе тестирования выявлено, что система требует дополнительной калибровки датчика влажности для разных типов почв, так как изначальные настройки могут давать погрешность в особо сухих или плотных грунтах. Это не критично для базового использования, но может потребовать ручной корректировки в нестандартных условиях.

Качество системы подтверждено серией тестов, включающих проверку аппаратной части, алгоритмов управления и пользовательского интерфейса.

Датчик влажности показал стабильную работу в стандартных условиях, но при высокой засоленности почвы или при наличии электромагнитных помех возможны ложные срабатывания.

Реле и помпа функционируют безотказно, однако при длительной непрерывной работе (более 5 минут) наблюдается перегрев помпы, что потребовало внедрения ограничения по времени работы.

Алгоритм автоматического полива корректно реагирует на изменения влажности, но при резких перепадах температуры (например, в теплицах) требуется дополнительная фильтрация данных для исключения ложных срабатываний.

Wi-Fi-модуль работает устойчиво, но в условиях слабого сигнала возможны задержки в передаче данных. Телеграмм-бот интуитивно понятен, но требует доработки, для большего удобства пользователей. Система демонстрирует высокую надежность в стандартных условиях эксплуатации, но имеет ряд ограничений, связанных с внешними факторами (тип почвы, температура, качество связи). Эти недостатки не являются критичными и могут быть устранены в будущих версиях.

Для дальнейшего развития проекта предлагаются следующие направления доработок:

- а) расширение функционала,
- б) добавление датчиков температуры и освещенности для комплексного контроля микроклимата,
- в) повышение надежности,
- г) использование более точных датчиков влажности с защитой от коррозии и помех,
- д) внедрение резервного питания (например, аккумулятора) для работы в условиях отключения электроэнергии,
- е) упрощение процесса первоначальной настройки (например, использование QR-кодов для быстрого подключения),
- ж) создание модульной архитектуры, позволяющей подключать несколько зон полива к одному контроллеру.

Проект автоматизированной системы полива можно считать успешным, так как он полностью соответствует заявленным требованиям и предоставляет удобное решение для ухода за растениями. Несмотря на выявленные мелкие недочеты, система готова к использованию в бытовых и сельскохозяйственных условиях. Дальнейшее развитие проекта позволит создать более универсальный и надежный продукт, востребованный на рынке умных технологий. Опыт, полученный в ходе разработки, показал важность тщательного тестирования в реальных условиях и учета внешних факторов, таких как тип почвы и качество связи. Эти знания будут полезны при реализации будущих проектов в области IoT и автоматизации [6].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Блум, Дж. Изучаем Arduino: инструменты и методы технического волшебства / Дж. Блум. – СПб.: Питер, 2020. – 336 с.
2. Шилдт, Г. C++: базовый курс / Г. Шилдт. – М.: Диалектика, 2021. – 624 с.
3. Петров, А.В. Автоматизированные системы полива в сельском хозяйстве: анализ современных решений / А.В. Петров, И.С. Козлов // Современные технологии автоматизации. – 2022. – № 4. – С. 45–52.
4. Официальная документация по ESP8266 и ESP32 [Электронный ресурс]. – URL: <https://docs.espressif.com/>.
5. Руководство по датчику влажности почвы Capacitive Soil Moisture Sensor v1.2 [Электронный ресурс]. – URL: <https://www.seeedstudio.com/>.
6. Форумы и сообщества по IoT (Stack Overflow, Arduino Forum) [Электронный ресурс]. – URL: <https://stackoverflow.com/>, <https://forum.arduino.cc/>.

ПРИЛОЖЕНИЕ А (обязательное)

Расчет энергопотребления системы

1. Состав системы и параметры компонентов

Для расчета возьмем следующие компоненты:

Микроконтроллер: ESP8266 (режим работы – активный + Wi-Fi)

Потребление в режиме активности: ~70 мА (при 3.3В)

Потребление в режиме глубокого сна (Deep Sleep): ~20 мкА

Датчик влажности почвы: Capacitive Soil Moisture Sensor

Потребляемый ток: ~5 мА (при 5В)

Водяная помпа: 12V DC, 5W

Потребляемый ток: ~420 мА (при 12В)

Релейный модуль (для управления помпой)

Потребление: ~70 мА (при 5В)

2. Сценарий работы системы

Каждые 30 минут (1800 секунд) просыпается, измеряет влажность (10 сек) и передает данные по Wi-Fi (5 сек).

Если влажность ниже порога – включает помпу на 15 секунд.

Затем снова переходит в Deep Sleep.

Цикл без полива (влажность в норме):

Активен: 15 секунд (10 сек датчик + 5 сек Wi-Fi)

Спит: 1785 секунд

Цикл с поливом:

Активен: 30 секунд (15 сек помпа + 10 сек датчик + 5 сек Wi-Fi)

Спит: 1770 секунд

3. Расчет среднего потребления

Ток в активном режиме (без полива):

ESP8266: 70 мА

Датчик: 5 мА

Итого: 75 мА

Ток в активном режиме (с поливом):

ESP8266: 70 мА

Датчик: 5 мА

Реле: 70 мА

Помпа: 420 мА (через отдельный блок питания, не учитываем для батареи ESP)

Итого: 145 мА (только для ESP и реле)

Средний ток за цикл (предположим, полив срабатывает 2 раза в сутки):

Без полива (22 цикла):

$$15 \text{ сек} \times 75 \text{ мА} + 1785 \text{ сек} \times 0.02 \text{ мА} = 1800 \text{ мА} \\ 1800 \text{ мА} \times 22 = 39600 \text{ мА}$$

С поливом (2 цикла):

$$30 \text{ сек} \times 145 \text{ мА} + 1770 \text{ сек} \times 0.02 \text{ мА} = 1800 \text{ мА} \\ 1800 \text{ мА} \times 22 = 39600 \text{ мА}$$

Среднее за сутки:

$$22 \times 0.63 + 2 \times 2.42 = 22.42 \text{ мА}$$

4. Оптимизация энергопотребления

Увеличение интервала между измерениями (например, до 1 часа):

Снижает средний ток до ~0.4 мА → время работы до 260 дней.

Отключение Wi-Fi (использовать только локальную логику):

Экономит ~50 мА в активном режиме → время работы до 200 дней.

Использование солнечной панели (5W, 12V):

Позволяет системе работать автономно неограниченно долго при хорошем освещении.