

ПРОГРАММНО-АППАРАТНАЯ СИСТЕМА “СЕЛЬСКОХОЗЯЙСТВЕННЫЙ ПОМОЩНИК”

Горунович Н.В., студент гр. 050501

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Перцев Д.Ю. – канд. техн. наук

В данной работе приведены основные технические решения и особенности создания программно-аппаратной системы “Сельскохозяйственный помощник”.

Разрабатываемая система представляет собой совокупность аппаратной части, представленной микроконтроллером, датчиками и реле, и программной части, представленной пользовательским интерфейсом, базой данных (БД) для хранения статистики и backend частью для взаимодействия пользователя и БД.

С микроконтроллером соединён WiFi модуль ESP-01 [1], поэтому взаимодействие датчиков и пользовательского устройства выполняется по протоколам IEEE 802.11b/g/n. Это обеспечивает бесперебойную работу на расстоянии от 140 до 250 метров при скорости от 11 до теоретически 600 Мбит/с. Для работы системы все устройства должны иметь доступ в Интернет.

Клиентская часть будет представлена Web приложением, что означает использование HTML, CSS и JS. В разработке используется библиотека React.js [2] со следующими подходами:

1. Компонентный подход, где пользовательский интерфейс разбивается на множество независимых блоков, отвечающих за отображение определенной части интерфейса.

2. Использование виртуального дерева (Virtual DOM, DOM), которое позволяет React эффективно обновлять только те части интерфейса, которые изменились.

3. Использование JavaScript XML (JSX) для описания пользовательского интерфейса. JSX позволяет объединять JavaScript и HTML в одном файле, что делает код более читаемым и логически разделённым.

4. Использование состояний: React позволяет компонентам иметь свои собственные состояния. Состояния используются для хранения и управления данными, которые могут изменяться при работе приложения.

Взаимодействие и обмен данными между датчиками и пользовательским устройством происходит в формате JSON, используя HTTP запросы GET и PUT. API разрабатывается с аппаратной стороны и является частью прошивки микроконтроллера.

Для управления БД используется PostgreSQL. Она работает по стандартам языка SQL, имеет высокую производительность, бесплатна для использования и поддерживает ряд функциональных возможностей, включая сложные типы данных, триггеры, транзакции.

Для разработки Backend части используется Java с фреймворками Spring [3]:

– Особенности фреймворка Spring Boot:

1. Автоматическая конфигурация и управление зависимостями.

2. Лёгкая настройка подключения к БД, указав соответствующие параметры в файле конфигурации.

3. Поддержка PostgreSQL.

– Особенности фреймворка Spring MVC:

1. Механизмы для обработки HTTP запросов и ответов в веб-приложениях.

2. Контроллеры Spring MVC могут взаимодействовать с базой данных через сервисы, которые в свою очередь используют репозитории или Data Access Object (DAO) для выполнения операций с данными.

– Особенности фреймворка Spring Data JPA:

1. Предоставляет абстракцию над Java Persistence API (JPA), упрощая работу с реляционными базами данных.

2. Позволяет создавать репозитории, которые автоматически создают SQL запросы на основе методов интерфейса, а также предоставляют удобные методы для выполнения операций CRUD (Create, Read, Update, Delete).

Список использованных источников:

1. ESP-01 WiFi Module [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.microchip.ua/wireless/esp01.pdf> – Дата доступа: 08.04.2024.

2. Документация React.js [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://react.dev/learn> – Дата доступа: 08.04.2024.

3. Документация Spring [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.spring.io/spring-framework/reference/index.html> – Дата доступа: 08.04.2024.