Ци Пэнжуй , Магистрант,
Амурский государственный университет

РАЗРАБОТКА И ВНЕДРЕНИЕ СИСТЕМЫ ТЕСТИРОВАНИЯ СВЕТОДИОДНОГО ОСВЕЩЕНИЯ

Аннотация. В данной главе представлен комплексный анализ процесса проектирования и реализации программного обеспечения многоканальной системы тестирования освещения. Система построена на модульной и многоуровневой архитектуре, включающей уровень аппаратной абстракции, сервисный уровень, уровень прикладной логики и интерфейсный уровень, что обеспечивает полный рабочий цикл от сбора данных, их обработки и локального отображения до удаленной передачи. В качестве интегрированной среды разработки выбран Keil C51 μVision, благодаря его оптимальной поддержке микроконтроллера STC89C52 и высокой эффективности компиляции. На этапе программирования аппаратного обеспечения реализованы ключевые алгоритмы, включая драйвер ADC0832, переключение каналов CD4051, цифровую фильтрацию и управление ЖК-дисплеем, а также обеспечена надежная интеграция программного и аппаратного обеспечения посредством холодного старта и системной отладки. Одновременно на основе Python разработаны облачный сервер и веб-интерфейс, реализующие функции удаленной визуализации данных и сетевой передачи. Финальное внедрение системы подтвердило соответствие проектным показателям по точности измерений (напряжение ±1%, ток ±2%), времени отклика (обнаружение неисправности <200 мс), синхронному мониторингу нескольких каналов и длительной стабильной работе. Система обладает хорошей расширяемостью и практической ценностью, предоставляя комплексное техническое решение для интеллектуального контроля систем освещения.

Ключевые слова: Архитектура программного обеспечения, Keil C51; отладка оборудования, STC89C52; сетевой интерфейс; развертывание системы.

Разработка программного обеспечения для многоканальной системы тестирования освещения осуществлялась с использованием модульного и структурированного подхода, что обеспечило высокую степень читаемости, удобства сопровождения и расширяемости кода. В качестве ядра системы был выбран микроконтроллер STC89C52, который отвечает за управление всеми аппаратными модулями, обработку данных, локальный вывод информации и взаимодействие с внешними системами. Программная архитектура разделена на несколько логических уровней, каждый из которых выполняет строго определенные функции и взаимодействует с другими уровнями через четко определенные интерфейсы.

Архитектура программного обеспечения

Программное обеспечение системы построено по многоуровневой архитектуре, включающей следующие основные слои:

Уровень аппаратной абстракции (HAL): Этот уровень отвечает за прямое взаимодействие с аппаратными компонентами, такими как микроконтроллер, АЦП (ADC0832), мультиплексор (CD4051), ЖК-дисплей (LCD1602) и модули связи. Он предоставляет вышележащим уровням унифицированные программные интерфейсы (API) для управления оборудованием, скрывая специфические детали реализации драйверов. Например, функции ADC_ReadChannel(uint8_t ch) или LCD_DisplayString(uint8_t line, char* str) позволяют верхним уровням работать с абстракциями, а не с регистрами микросхем.

Уровень служб (Service Layer): На этом уровне реализованы базовые сервисы, необходимые для функционирования системы: менеджер прерываний, диспетчер задач,

модуль обработки данных (фильтрация, калибровка), модуль управления питанием и таймеры реального времени. Эти сервисы используются бизнес-логикой для выполнения периодических или событийно-управляемых операций.

При разработке встраиваемых систем выбор подходящей среды разработки программного обеспечения имеет решающее значение для успеха проекта. В данной разработке программного обеспечения многоканальной системы тестирования освещения, основываясь на комплексном рассмотрении эффективности разработки, качества кода, удобства отладки и экономической эффективности, в качестве основной интегрированной среды разработки (IDE) был выбран Keil C51 μVision. Keil C51 – это зрелый инструментарий разработки на языке C, специально разработанный немецкой компанией Keil (ныне входящей в Arm) для семейства микроконтроллеров 8051 и их совместимых вариантов (таких как STC89C52), получивший широкое признание в промышленности и академических кругах.

Конкретные причины выбора Keil C51

Превосходная производительность компилятора:

Высокая эффективность кода: Компилятор Keil C51 известен своей выдающейся эффективностью генерации кода. Он способен выполнять глубокую оптимизацию для архитектуры 8051, создавая компактный и быстрый машинный код. Это особенно важно для систем с ограниченными ресурсами памяти (у STC89C52 всего 8 КБ Flash и 512 Б ОЗУ), гарантируя плавную работу всех функциональных модулей (сбор данных, обработка, отображение, связь) на ограниченном аппаратном обеспечении.

Для удалённого мониторинга и визуализации данных в систему добавлены облачный сервер на базе Python и веб-интерфейс. Этот функциональный модуль, являясь важным расширением системы, реализует полноценный канал передачи данных от полевых устройств к облаку, а затем к браузеру пользователя.

Архитектура системы и поток данных

Терминальный уровень (STC89C52) - микроконтроллер отвечает за сбор данных и их передачу на Wi-Fi-модуль (например, ESP8266) через UART.

Сетевой уровень передачи (Wi-Fi-модуль) – Wi-Fi-модуль настроен на прозрачный режим передачи, подключается к локальному маршрутизатору и устанавливает соединение TCP-сокет с сервером Python. Он прозрачно пересылает последовательные данные, отправляемые микроконтроллером, на указанный IP-адрес и порт сервера.

Серверный уровень (бэкенд Python) - бэкенд-сервис, написанный на Python, работает на облачном сервере с публичным IP-адресом или на локальном компьютере. Эта служба в основном выполняет следующие задачи:

Сервер сокетов – прослушивает определенный порт и принимает TCP-соединения с различными терминалами тестирования освещения.

Анализ и обработка данных - анализирует кадры данных с микроконтроллера в соответствии с пользовательским протоколом связи и проверяет целостность данных (например, с помощью CRC-контроллера).

Хранение данных – полученные исторические данные постоянно хранятся в базах данных (например, MySQL, SQLite или InfluxDB) для удобства запросов и анализа.

Веб-API – интерфейсы RESTful API создаются с использованием веб-фреймворков, таких как Flask или FastAPI, для предоставления сервисов запросов данных к веб-страницам front-end.

Список литературы:

1. PwC. (2023). Sports Industry: Outlook and Trends 2023.

2 "Anthony Joshua and the Tech Behind the Champ." (2022). ESPN The Magazine.

3. Bailenson, J. (2023). Infinite Reality: The Hidden Blueprint of Our Virtual Lives. William Morrow.

4. USOPC. (2022). Health Monitoring Guidelines for High-Performance Athletes.

5. International Boxing Association. (2023). Global Boxing Development Program Annual Report.