

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»
Институт информационных технологий



ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

**59-я научная конференция
аспирантов, магистрантов и студентов**

Сборник статей

17–21 апреля 2023 года
Минск, ИИТ БГУИР

УДК 004+ 621.3
ББК 32.97+32.81

Информационные системы и технологии : сборник статей 59-й научной конференции аспирантов, магистрантов и студентов (г. Минск, 17 - 21 апреля 2023 г.) / Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники; редкол. : А. И. Парамонов [и др.] – Минск, 2023. – 81 с.; ил.

В сборнике представлены статьи по материалам 59-й научной конференции аспирантов, магистрантов и студентов БГУИР.

Для научных и инженерно-технических работников, преподавателей, аспирантов, магистрантов и студентов вузов.

Редакционная коллегия:

А.И. Парамонов, кандидат технических наук, доцент, заведующий кафедрой информационных систем и технологий ИИТ БГУИР (председатель секции «Программная и компьютерная инженерия»).

И.П. Сидорчук, заместитель директора ИИТ по научно-воспитательной работе.

М.Л. Маковский, заместитель декана ФКТ ИИТ БГУИР (председатель комиссии по организации и проведению конференции «Информационные системы и технологии»).

С.В. Власюк, ведущий специалист ФКТ ИИТ БГУИР (заместитель председателя комиссии по организации и проведению конференции «Информационные системы и технологии»).

О.Ю. Кунцевич, кандидат педагогических наук, доцент, доцент кафедры информационных систем и технологий ИИТ БГУИР.

А.Г. Савенко, магистр технических наук, старший преподаватель кафедры информационных систем и технологий ИИТ БГУИР.

О.А. Шведова, магистр технических наук, старший преподаватель кафедры информационных систем и технологий ИИТ БГУИР.

В.А. Сицко, магистр технических наук, старший преподаватель кафедры информационных систем и технологий ИИТ БГУИР.

А.В. Матвеев, старший преподаватель кафедры информационных систем и технологий ИИТ БГУИР.

Ответственность за достоверность публикуемых материалов несут их авторы.

© УО «Белорусский государственный университет информатики и радиоэлектроники», Институт информационных технологий, 2023

СОДЕРЖАНИЕ

1	ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА СТОМАТОЛОГИЧЕСКИХ УСЛУГ В ПОЛИКЛИНИКЕ.....	5
2	УМНАЯ СИСТЕМА ХРАНЕНИЯ И УЧЕТА РАДИОДЕТАЛЕЙ И КОМПЛЕКТУЮЩИХ.....	8
3	ПРОГРАММНОЕ СРЕДСТВО АНАЛИЗА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ	13
4	ЗАРЯДНОЕ УСТРОЙСТВО С ИСПОЛЬЗОВАНИЕМ СОЛНЕЧНОЙ ЭНЕРГИИ.....	16
5	АВТОМАТИЗИРОВАННАЯ СИСТЕМА СЦЕНИЧЕСКОГО СОПРОВОЖДЕНИЯ ВЫСТУПЛЕНИЙ	20
6	ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ ВЕРСТКИ И ПОДГОТОВКИ К ПЕЧАТИ МАТЕРИАЛОВ ДЛЯ НАСТОЛЬНЫХ ИГР.....	25
7	ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ МЕДИЦИНСКОГО ЦЕНТРА.....	31
8	АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ПОКРАСОЧНОЙ КАМЕРОЙ.....	35
9	ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА РАБОЧИХ КАДРОВ ПРЕДПРИЯТИЯ.....	39
10	АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПЕРВИЧНОЙ ОБРАБОТКИ РАДИОЛОКАЦИОННОЙ ИНФОРМАЦИИ.....	42
11	РАСПРЕДЕЛЕННАЯ СИСТЕМА СТАТИСТИЧЕСКОГО АНАЛИЗА ДОКУМЕНТОВ.....	45
12	ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ ПРИЕМНОЙ КОМИССИИ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ.....	48
13	АППАРАТНОЕ ОБЕСПЕЧЕНИЕ МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ.....	53

14	МОДЕЛИ ОПРЕДЕЛЕНИЯ РАБОТОСПОСОБНОСТИ МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ.....	56
15	МОДЕЛИ СИСТЕМ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ ПАРАМЕТРОВ ОБЪЕКТОВ И ПРОЦЕССОВ.....	60
16	ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ.....	64
17	ВЕБ-ПРИЛОЖЕНИЕ «ВЫПУСКАЮЩАЯ КАФЕДРА» ДЛЯ АВТОМАТИЗАЦИИ И ЦИФРОВИЗАЦИИ ДИПЛОМНОГО ПРОЕКТИРОВАНИЯ.....	68
18	ТРАНСФОРМАЦИЯ ПРИЛОЖЕНИЙ ОТ МОНОЛИТНОЙ К МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ.....	73
19	СИСТЕМА УПРАВЛЕНИЯ МИКРОКЛИМАТОМ ПРИ АККЛИМАТИЗАЦИИ РАСТЕНИЙ.....	76

УДК 004.42+616.314

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА СТОМАТОЛОГИЧЕСКИХ УСЛУГ В ПОЛИКЛИНИКЕ

Бич М.В. студент

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Кунцевич О.Ю. – канд. пед. наук, доцент каф. ИСиТ

Аннотация. Рассмотрены вопросы разработки программного средства для учета стоматологических услуг, представлен концепт пользовательского интерфейса приложения, обоснован выбор средств разработки, перечислены достоинства и недостатки применения разработки на основе .NET Windows Forms. В качестве языка программирования выбран язык C#. Для хранения информации была создана база данных в СУБД SQL Server.

Ключевые слова. Программное обеспечение, разработка приложений, стоматологическая поликлиника, услуга, автоматизация учета.

Введение. За последние несколько лет быстрыми темпами развиваются информационные технологии, происходит их дальнейшее внедрение в различные сферы деятельности общества. Так, в частности, расширение бизнеса требует четкого и своевременного контроля, который можно получить, автоматизируя услуги с помощью программных средств.

В настоящее время ситуация на рынке стоматологических услуг складывается таким образом, что конкуренция очень высока. Поэтому для стабильной и доходной работы необходимо укреплять свои позиции на рынке и усилить сферу влияния фирмы [1].

Автоматизация работы стоматологических поликлиник с помощью программных средств влияет на эффективность управленческой деятельности, облегчает ведение учета пациентов, увеличивает скорость выполнения различных операций, способствует сокращению времени на оформление документации, повышению экономических показателей организации и успешности предприятия в целом.

Стоматологическая поликлиника оказывает все виды соответствующей лечебной, диагностической и консультационной помощи.

Отсутствие автоматизированного процесса учета пациентов может отрицательно сказаться на качестве предоставляемых услуг. Поэтому существует объективная необходимость в разработке соответствующих программных средств.

Целью нашего проекта является разработка программного средства Dental Clinic для автоматизации учета стоматологических услуг в поликлинике.

Задачи, способствующие достижению цели:

— разработка программного средства, которое предоставит возможность учета стоматологических услуг в поликлинике.

— создание административного модуля управления разработанным приложением, который позволит администратору получить доступ к базе данных и предоставит возможность добавлять, удалять, редактировать записи об услугах, пациентах и сотрудниках поликлиники.

— проектирование простого, удобного, интуитивно понятного рабочего интерфейса, ориентированного на применение пользователем с любым уровнем технической подготовки.

Для реализации проекта необходимо:

— обосновать необходимость создания программного средства.

— провести сравнения программных средств и выбрать среду разработки, основной функционал и компоненты управления, а также язык программирования.

— разработать программное средство для автоматизации учета рабочих кадров предприятия на платформе .Net Windows Forms.

— провести тестирование разработанного программного средства.

— разработать руководство пользователя.

Объектом исследования является стоматологическая поликлиника.

Предметом исследования является автоматизация деятельности стоматологической поликлиники, в частности процессов учета пациентов, услуг и деятельности сотрудников. Учет пациентов включает в себя хранение личных данных, информацию об оказанных услугах, их стоимость и данные персонала.

Основная часть. В начале разработки нами был проведен анализ предметной области, а также рассмотрены конкурирующие системы (таблица 1). Выявленные недостатки аналогов было решено устранить в собственной разработке.

Таблица 1 – Аналоги разрабатываемого программного средства [2-4]

ProfmanDental	StomX	Dental4Windows
<ul style="list-style-type: none"> — платная версия, есть бесплатная демоверсия на 14 дней; — без интернета не работает. 	<ul style="list-style-type: none"> — полнофункциональная версия является платной. — не работает без интернета. 	<ul style="list-style-type: none"> — устаревший дизайн интерфейса. — высокая стоимость. — высокие требования к конфигурации пользовательских ПК.

Далее были определены назначение и возможности разработанного программного средства (ПС), в частности, упрощение ведения учета пациентов и автоматизация учета стоматологических услуг в поликлинике. Также программное средство будет выполнять следующие основные функции:

- добавление, изменение и удаление данных в таблицах «Сотрудники»;
- добавление, изменение и удаление данных в таблице «Услуги»;
- добавление, изменение и удаление данных в таблице «Пациенты»;
- экспорт данных в Excel;
- поиск данных в таблицах.

Для реализации поставленных задач были выбраны следующие средства разработки: Microsoft Visual Studio с использованием .NET Windows Forms, язык программирования – C#, СУБД – SQL Server, среда разработки Management Studio, графические материалы оформлялись, в частности, с помощью Microsoft Visio.

Одним из предпроектных этапов разработки программного обеспечения является этап системного анализа и моделирования предметной области. Для реализации существуют различные CASE-средства, позволяющие осуществлять построение соответствующих моделей и диаграмм. Так, нами построена контекстная диаграмма IDEF0 (рисунок 1), а также диаграмма декомпозиции первого уровня.

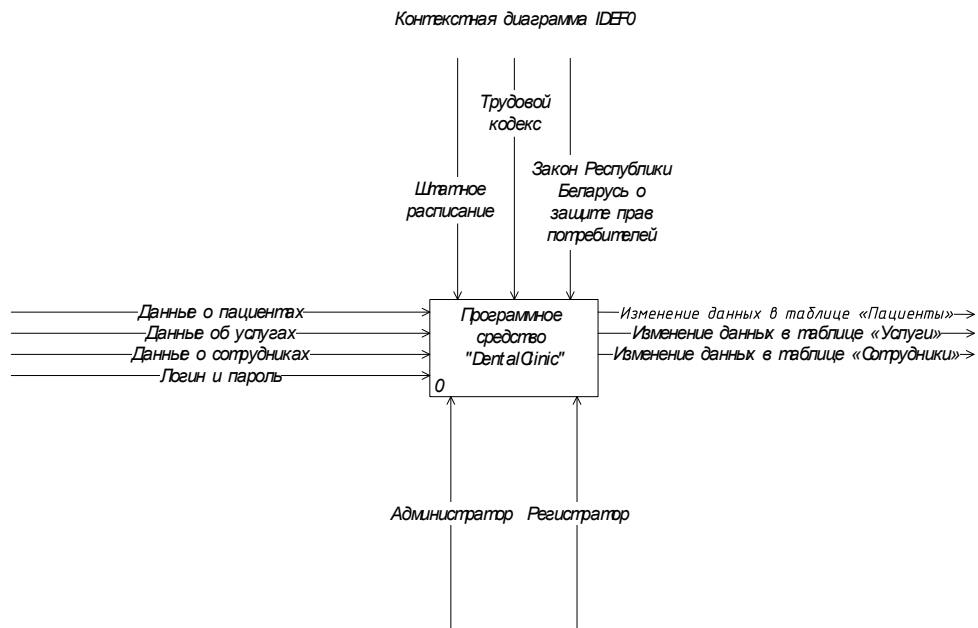


Рисунок 1 – Контекстная диаграмма IDEF0

Программное средство DentalClinic состоит из трех основных форм: услуги, пациенты и сотрудники.

С помощью формы *Сотрудники* и *Услуги* возможно осуществить поиск данных в соответствующих таблицах. С помощью формы *Пациенты* (рисунок 2) возможно осуществить поиск данных в таблице, а также экспортировать данные в Excel, добавить и изменить информацию о пациентах.

Для описания механизмов работы модулей программы были разработаны, в частности, алгоритм программы обработки данных, алгоритм работы программного средства, диаграмма вариантов использования, база данных др. Так, например, в диаграмме деятельности описан рабочий процесс с помощью следующих шагов:

- для регистратора: просмотр и поиск данных в таблицах «Сотрудники», «Услуги» и «Пациенты». Из таблицы «Пациенты» возможно экспортировать данные в Excel, выполнять

заполнение таблицы данными, выбирать сотрудника и услуги. После всех манипуляций необходимо сохранить данные;

— для администратора: просмотр данных в таблице «Сотрудники», «Услуги» и «Пациенты», а также обработка данных (добавление, удаление, изменение), экспорт данных в Excel из таблицы «Пациенты».

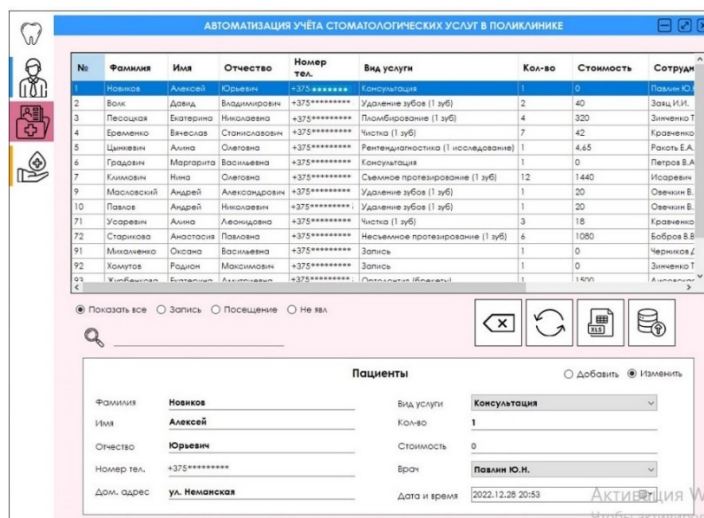


Рисунок 2 – Форма «Пациенты»

Заключение. Разработанное приложение позволит пользователю-регистратору записать пациента на прием, заполнить данные о пациенте и об оказанных услугах, указать их стоимость и данные персонала. А у администратора есть доступ для изменения данных во всех таблицах: «Пациенты», «Сотрудники» и «Услуги». Основные уникальные функции, отличающие средство от аналогов, заключаются в доступности и простоте пользования приложением, задействовании небольшого объема памяти компьютера, современном дизайне, высокой скорости работы программы, отсутствии обязательной регистрации для работы с приложением.

Список использованных источников:

1. *Маркетинговая стратегия частной стоматологической клиники [Электронный ресурс]. – Режим доступа: <https://works.doklad.ru/view/58jfb1ChrMM.html>.*
2. *ProfmanDental: программа по управлению частной стоматологией [Электронный ресурс]. – Режим доступа: <https://profmandental.by/>.*
3. *StomX: программа для стоматологии [Электронный ресурс]. – Режим доступа: <https://stomx.ru/>*
4. *Dental4Windows: программа для стоматологии [Электронный ресурс]. – Режим доступа: <https://d4w.ru/>.*

UDC 004.42+616.314

SOFTWARE TOOL FOR AUTOMATION OF ACCOUNT OF DENTAL SERVICES IN A POLYCLINIC

Bich M.A.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Kuntsevich O.Yu. – Candidate of Pedagogic Sciences, Associate Professor

Annotation. The article discusses the development of a software tool for accounting for dental services, presents the concept of the user interface of the application, justifies the choice of development tools, lists the advantages and disadvantages of using development in the form of .NET Windows Forms. C# was chosen as the programming language. To store information, a database was created in the SQL Server DBMS.

Keywords. Software, application development, dental clinic, service, accounting automation.

УДК 004.42+004.522

УМНАЯ СИСТЕМА ХРАНЕНИЯ И УЧЕТА РАДИОДЕТАЛЕЙ И КОМПЛЕКТУЮЩИХ

Варченя И.В., студент, Савенко А.Г., маг. техн. наук

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

Савенко А.Г. – маг. техн. наук, ст. препод. каф. ИСиТ

Аннотация. В данной работе представлен программно-аппаратный комплекс для автоматизации и интеллектуализации хранения и складского учета радиодеталей и комплектующих на предприятии, выполненный на аппаратной платформе Raspberry Pi с голосовым управлением и визуальной светоиндикацией обработки запросов. Разработанная программно-аппаратная система хранения и складского учета радиодеталей для предприятий может быть легко масштабирована и адаптирована под специфику и нужды различных предприятий и позволит автоматизировать и облегчить этот процесс.

Ключевые слова. Raspberry Pi, голосовые запросы, световая индикация, складской учет, программно-аппаратный комплекс.

Введение. В современном мире существует множество решений, автоматизирующих работу на производстве, в сельском хозяйстве и быту. Все подобные решения управляются с помощью определенного перечня команд, скрытых от рядового пользователя под интерфейсами разнообразных мобильных и компьютерных приложений.

В настоящей работе представлен разработанный проект по автоматизации и интеллектуализации системы хранения и складского учета деталей и комплектующих, включающий в себя разработанное программно-аппаратное обеспечение и прототип системы. Прототип предлагаемой системы предназначен для хранения и складского учета радиодеталей на предприятии, занимающемся ремонтом электрооборудования. Сама система легко масштабируется за счет соответствующего изменения аппаратной части системы (соответствующих задаче датчиков, сенсоров и т.д.).

Основная часть. Умная система хранения и складского учета реализована на базе микрокомпьютера Raspberry Pi, что позволяет экономно и эргономично выстроить структуру физической части проекта, для управления системой. Raspberry Pi – это одноплатный микрокомпьютер, изначально разработанный как бюджетная система для обучения информатике, впоследствии получивший намного более широкое применение и популярность, чем ожидали его разработчики [1].

К микрокомпьютеру Raspberry Pi (рис. 1 а) подключаются периферийные устройства, такие как датчики веса (для определения наличия и количества хранимых деталей) (рис. 1 б), светодиодные устройства индикации (для визуализации найденных контейнеров для хранения или выводе ошибки поиска), микрофон (для приема поступающих голосовых команд) и другие устройства, в зависимости от специфики системы хранения.

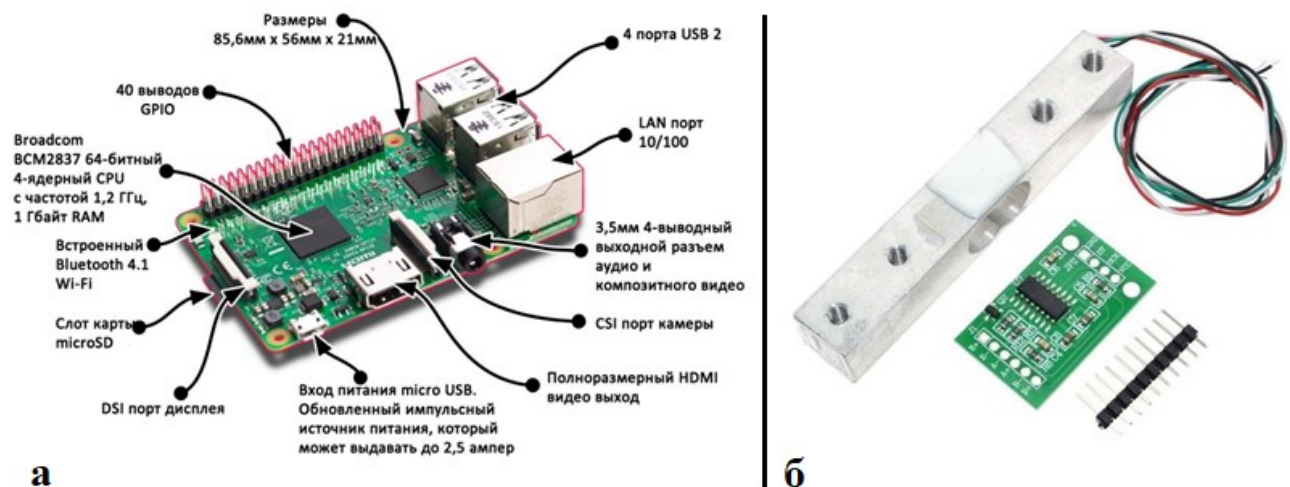


Рис.1 – Микрокомпьютер Raspberry Pi (а) и тензодатчик и драйвер на микросхеме HX711 (б)

Тензодатчик и микросхема HX711 – связка, которая позволит создавать устройства для измерения веса или давления, оказываемого на поверхность датчика, а затем передавать эти показания (с высокой точностью) на плату микрокомпьютера.

Для световой индикации результатов поиска (определённого стеллажа или контейнера) по голосовому поисковому запросу (зелёным цветом), а также для индикации отсутствия результатов поиска (красным цветом) предлагается использовать светодиодные устройства индикации. Пример схемы подключения светодиодных устройств индикации для стеллажа с размещенными на нем шестнадцатью контейнерами для хранения (разработанная физическая модель прототипа) приведен на рисунке 2.

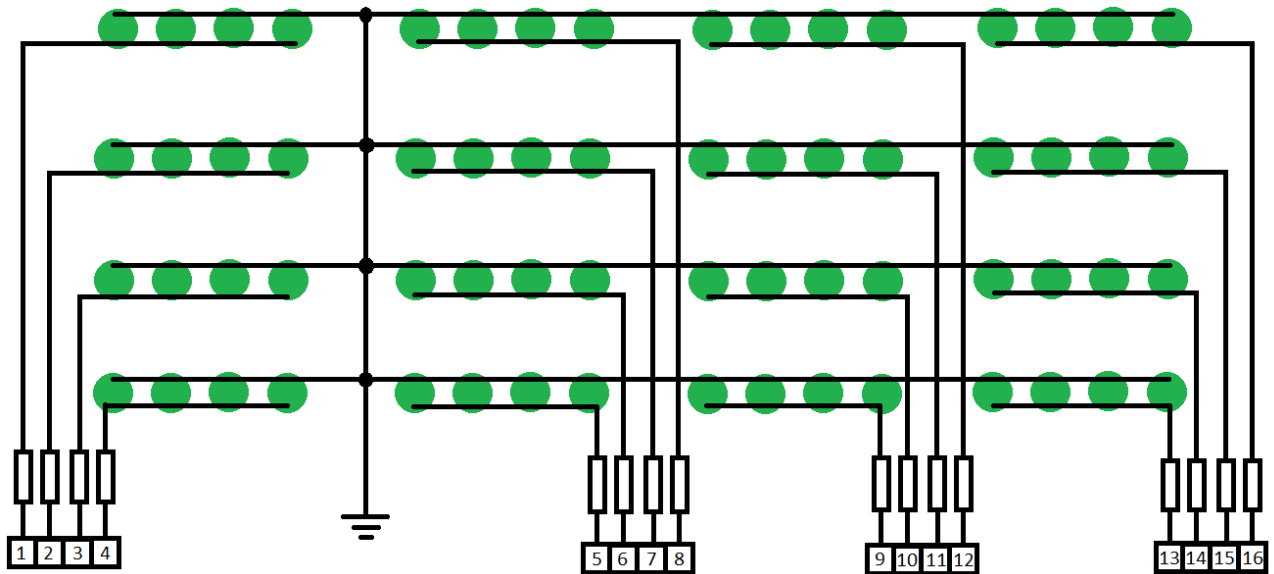


Рис.2 – Пример подключения светодиодной индикации стеллажа

Для программного управления системой разработано программное средство, реализующее управление посредством голосового интерфейса, и не требующее специальной подготовки сотрудников. Разработанное программное средство использует ряд технологий, реализующих следующие функции:

- обработка голосовых команд;
- обработка информации, считываемой с датчика веса;
- обработка информации о размещенных деталях;
- вывод данных посредством светодиодной индикации.
- Данный функционал решает следующие задачи:
 - голосовое управление программным средством;
 - хранение информации о размещенных деталях;
 - ввод данных датчика веса;
 - вывод данных на индикаторные светодиоды;
 - учет режимов работы с физической моделью.

Общий алгоритм работы программного средства управления системой предполагает получение команд поиска хранящихся деталей, добавления деталей на хранение, изъятие (удаление) деталей со складского хранения. После запуска программы на физической модели системы загорается светодиод, который служит индикатором работы программного средства. Далее пользователю необходимо произнести вслух команду управления.

Команды управления реализуются следующим образом: после считывания микрофоном команды, голос преобразуется в текст. Голосовые команды поступают в блок ввода голосовых команд. На этом этапе голосовые команды представляют вибрации воздушной среды, воспринимаемые микрофоном, которые производит пользователь. Далее сигнал поступает в блок анализа полученных сигналов с микрофона. После текстовая строка разбивается на составляющие в блоке обработки голосовых команд, согласно списку возможных голосовых команд, и выдается команда об изменении записи в файле хранения. Если команда распознана, то запускается соответствующий метод обработки: поиск, удаление, добавление и изменение количества деталей. Если команда не распознана, то включается индикация об ошибке. Информация с датчика веса анализируется, а далее задается команда индикации активации в соответствующем блоке. Команда индикации формирует команду индикацию об ошибке или индикацию заданной ячейки в соответствующих блоках. Далее команды индикации активации, ошибки и заданной ячейки поступают в блок передачи команды микроконтроллеру, который формирует управляющие сигналы на портах ввода-вывода микроконтроллера. Программа завершается прерыванием пользователя.

Диаграмма вариантов использования программного средства представлена на рисунке 3.

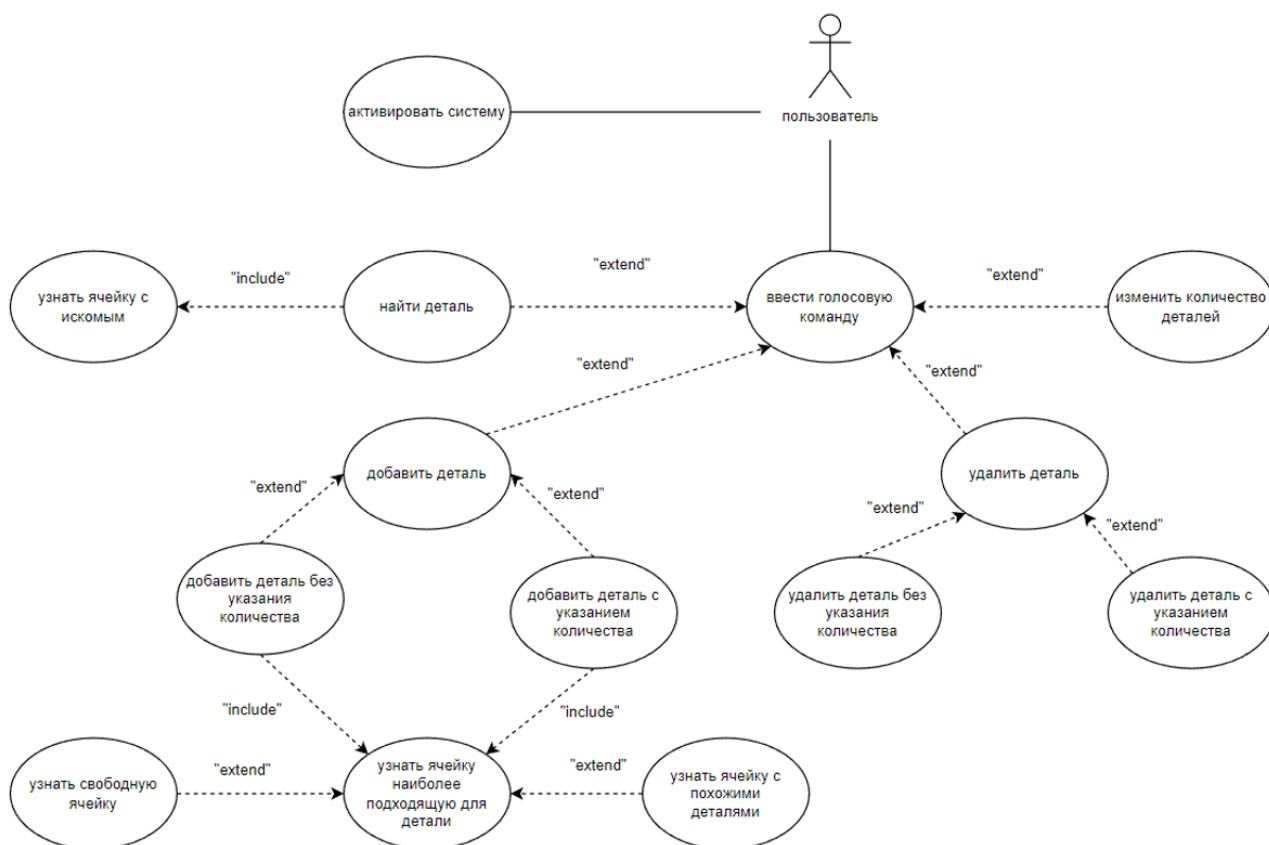


Рис.3 – Диаграмма вариантов использования программного средства

В случае поиск детали может быть два исхода: деталь найдена и деталь не обнаружена. В первом случае будет подсвечиваться необходимая ячейка при помощи индикаторных светодиодов. Во втором – отобразится индикация об ошибке.

В случае команды добавления новой детали – необходимая (с уже аналогичными деталями или, если таких нет, пустая ячейка подсветится. Команда добавления детали может быть двух видов: с указанием количества и без указания количества (тогда количество считается равным единице).

Команда удаления детали также может быть с и без указания количества радиодеталей. Например: «Удалить вольтметр». Или: «Удалить десять реле».

При вводе команды изменения количества деталей производится проверка наличия похожих деталей. Если проверка пройдена, то подсветится ячейка с похожими деталями, если нет, то подсветится пустая ячейка. Команда удаления должна быть с указанием количества удаляемых деталей. В случае команды удаления радиодетали будет отображена соответствующая индикация удаления детали из определённой ячейки и будет выполнено изменение их количества.

В ходе разработки программного средства были разработаны следующие модули, иерархическая модель которых представлена на рисунке 4.

Работа приложения начинается после запуска модуля управления программным средством, в котором находится код, реализующий основной алгоритм, посредством вызовов методов из других модулей.

Модуль констант хранит в себе неизменяемые параметры с информацией о названии файла хранения информации, значениями номеров управляющих светодиодами выводов, значениями номера выводов для обработки информации с датчика веса, строковые значения для работы с голосовыми командами.

Модуль считывания команд представляет набор методов, который отвечает за считывание голосовых команд непосредственно через микрофон и преобразование голосовых команд в текстовые строки.

Модуль парсинга текста представляет набор методов, который преобразует текстовые строки в понятный для программы набор параметров. А также подготавливает информацию для записи в Excel файл для хранения.

Модуль управления электроникой включает два основных блока кода: блок управления светодиодами и блок считывания показаний с датчика веса.

Блок управления светодиодами содержит методы для отображения определённой ячейки, а также для отображения индикации об ошибке и удалении радиодетали из ячейки.

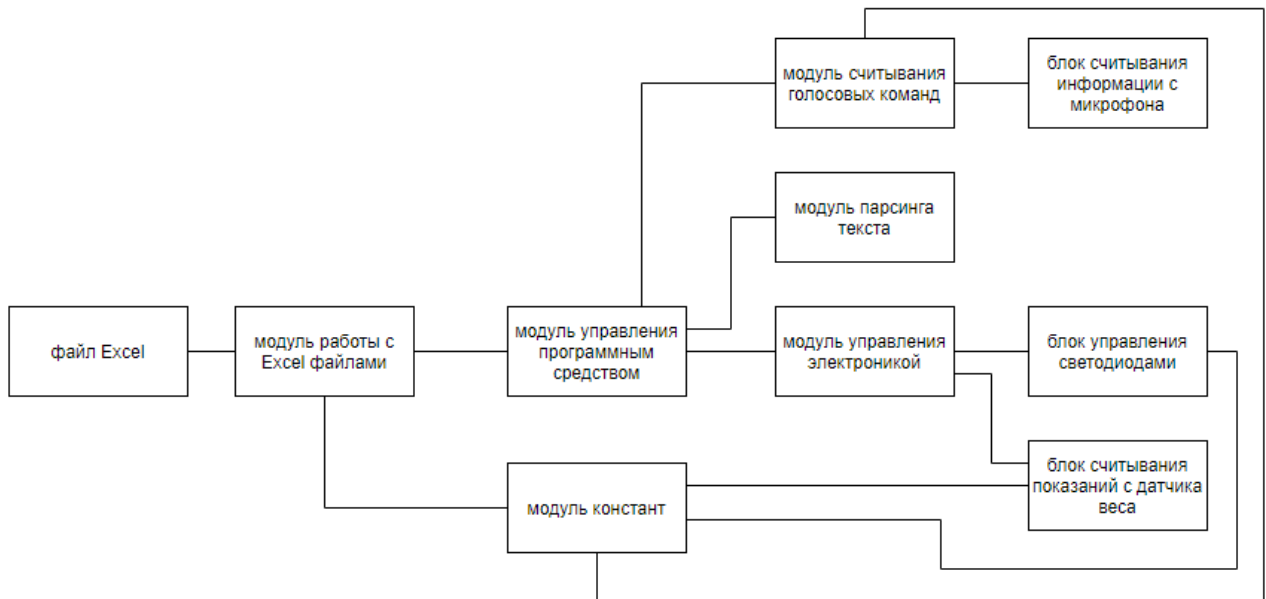


Рис. 4 – Структура программного средства

Блок считывания показаний с датчика веса содержит методы для отслеживания изменения веса системы, а также метод для обработки нажатий пользователя, для активации системы.

Модуль работы с Excel-файлами содержит методы для работы с хранимой информацией. Это методы поиска, добавления, удаления и изменения количества деталей.

Для корректной работы приложения, а также для наполнения его информацией используется также файл хранения, который представляет собой Excel-файл состоящая из одной таблицы размерами N на M ячейки, что соответствует размерам физической модели стеллажа хранения, где N – количество полок стеллажа, M – количество ячеек (контейнеров) на полке стеллажа.

В каждой ячейке таблицы Excel-файла может быть неограниченное количество записей. Каждая запись представляет собой структуру из наименования хранимых деталей, двоеточия и знака пробела, числа, указывающего количество радиодеталей. Если в ячейке таблицы находится несколько записей, то они разделены знаком переноса строки. Все это позволяет использовать файл не только посредством голосового управления программным средством, но также и в ручном режиме. Пример таблицы Excel-файла представлен на рисунке 5.

	A	B	C	D
1	светодиоды: 6 лампочки: 3			
2		трансформаторы: 1 дроссели: 2		
3				фоторезисторы: 5
4			датчик движения: 2	

Рис. 5 – Пример таблица хранения информации о количестве деталей

Физическая модель созданного прототипа умной системы хранения и складского учета разработана для хранения радиодеталей и представлена на рисунке 6.



Рис. 6 – Прототип разработанной системы

Заключение. Таким образом, разработана программно-аппаратная умная система хранения и складского учета радиодеталей для предприятий, которая легко может быть масштабирована и адаптирована под специфику и нужды различных предприятий и позволит автоматизировать и облегчить этот процесс.

Список использованных источников:

1. 3d-diy.ru [Электронный ресурс]. Режим доступа: <https://3d-diy.ru/wiki/arduino-platy/obzor-plat-raspberry-pi/>. Дата доступа: 15.03.2023.

UDC 004.42+004.522

SMART SYSTEM OF STORAGE AND ACCOUNTING OF RADIO PARTS AND COMPONENTS

Varchenya I.V., Savenko A.G.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Savenko A.G. – Master Of Engineering Sciences

Annotation. This paper presents a software and hardware system for automating and intellectualizing the storage and inventory control of radio components and components in an enterprise, implemented on the Raspberry Pi hardware platform with voice control and visual LED indication of request processing. The developed software and hardware system for storage and inventory accounting of radio components for enterprises can be easily scaled and adapted to the specifics and needs of various enterprises and will automate and facilitate this process.

Keywords. Raspberry Pi, voice requests, light indication, inventory control, hardware and software system.

УДК 621.396.6

ПРОГРАММНОЕ СРЕДСТВО АНАЛИЗА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ

Величко В.М., студент, Скудняков Ю.А, канд. техн. наук

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В работе разработано программное средство для осуществления сравнительного анализа эффективности алгоритмов размещения элементов электрической схемы на печатной плате при вариации их количества с целью достижения минимальной суммарной длины связей между ними, что позволяет снизить трудоемкость, затраты производства конструкций изделий, повысить их надежность и долговечность.

Ключевые слова. Программное средство, эффективность, алгоритмы, размещение, элементы, электрическая схема.

Введение. Конструирование электронных средств производится в течение трех этапов:

1) компоновки элементов разрабатываемых электронных изделий, когда производится распределение элементов низшего уровня иерархии по узлам высшего уровня, при этом, решаются две задачи: покрытие – преобразование исходной схемы в схему соединения элементов, номенклатура которых задана и разрезание – разбиение исходной схемы на части;

2) размещения конструктивных модулей электрической схемы изделия, основной целью которого является создание наилучших условий для последующей трассировки соединений при удовлетворении основных требований, обеспечивающих работоспособность проектируемой схемы;

3) трассировки электрических соединений между всеми элементами печатной платы проектируемого электронного изделия, при этом необходимо учитывать различные конструктивно-технические ограничения: допускаются пересечения трасс или нет, возможен ли переход с одного слоя на другой, сколько слоев отводится для трассировки, допустимые ширина проводников и расстояния между ними и т. д., а применяемые алгоритмы трассировки существенно зависят от принятой конструкции и технологии изготовления электронной аппаратуры [1,2].

В целом критерии размещения элементов электрической схемы целесообразно объединить в 3 группы:

1) конструкторско-технологические (минимальная суммарная длина связей между элементами электрической схемы; минимальное число и длина самых протяженных связей схемы);

2) электромагнитной совместимости элементов электрической схемы (минимальное число паразитных связей между элементами; минимальный уровень паразитных наводок);

3) тепловой совместимости элементов электрической схемы (минимальное тепловое влияние наиболее нагретых элементов на другие, особенно на соседние; равномерное распределение температуры в монтажном пространстве).

Рассматриваемая работа посвящена исследованию и практическому применению алгоритмов парных и групповых перестановок элементов электрической схемы при их размещении на печатной плате, при этом, под размещением понимается процесс определения оптимального пространственного расположения конструктивных модулей разрабатываемого электронного средства в монтажном пространстве. Рассматриваемые алгоритмы относятся к итерационным, использование которых позволяет выполнять перераспределение элементов электрической схемы для уменьшения общей длины соединений между ними. Каждый этап алгоритма парных или групповых перестановок должен приводить к новому размещению элементов схемы с меньшей общей суммарной длиной. Такие алгоритмы относятся к полиномиальным со сложностью порядка $O(n^2) - O(n^4)$, где n – число элементов электрической схемы.

Целью данной работы является сравнение эффективности применения вышеприведенных алгоритмов размещения при минимизации суммарной длины связей между всеми элементами схемы.

Для достижения сформулированной цели разработано программное средство (ПС), реализующее алгоритмы парных и групповых перестановок.

В дальнейшем планируется осуществить разработку ПС для решения задач размещения на основе венгерского алгоритма, алгоритма силонаправленной релаксации и алгоритма на основе метода ветвей и границ. Для полноценного конструирования электронного средства желательно совместное решение задач размещения и трассировки. Однако, использование такого подхода вызывает проблемы его реализации в силу наличия сложных взаимосвязей между отдельными параметрами конструкции и

схем соединений. Поэтому программно-алгоритмическое решение указанных задач выполняется раздельно: сначала осуществляется процесс размещения, а затем – трассировки.

Основная часть. В работе для автоматизации решения поставленной задачи исходная принципиальная электрическая схема электронного средства представляется в виде графовой модели [3-5].

Рассмотрим математическую модель задачи размещения, которая представляется в виде: множества элементов электрической схемы $E = \{e_1, e_2, \dots, e_m\}$, $|E| = m$ и множества позиций монтажной плоскости $P = \{p_1, p_2, \dots, p_n\}$, $|P| = n$, в общем случае $n \geq m$.

Оптимальное решение задачи размещения может быть получено согласно универсальному критерию минимизации суммарной длины соединений между элементами электрической схемы:

$$F(P) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m r_{ij} \times d_{p(i)p(j)}. \quad (1)$$

В приведенной формуле: $F(P)$ – суммарная длина соединений; m – количество размещаемых элементов; r_{ij} – элемент матрицы связности $R = [r_{ij}]_{m \times m}$; $d_{p(i)p(j)}$ – элемент матрицы расстояний $D = [d_{ij}]_{n \times n}$.

В ортогональной метрике расстояние между элементами e_i и e_j вычисляется по формуле:

$$d_{ij} = |x_i - x_j| + |y_i - y_j|, \quad (2)$$

где $(x_i, y_i), (x_j, y_j)$ координаты позиций, а в евклидовой метрике (диагонализация трасс):

$$d_{ij} = ((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}, \quad (3)$$

где $(x_i, y_i), (x_j, y_j)$ координаты позиций.

В общем случае в качестве монтажной плоскости могут использоваться печатная плата, панель с проводными соединениями, подложка микросборки, кристалл БИС и т.д. В данной работе в качестве монтажного поля применяется печатная плата [6].

Сервисное решение использования разработанного ПС заключается в следующем.

Есть 3 поля и кнопка: количество элементов, размеры платы y и x и кнопка «Анализ» (рисунок 1).

После заполнения полей и нажатии кнопки «Анализ», составляется матрица связности графа и матрица со случайной длиной ребер графа. Далее происходит выполнение алгоритмов и вывода графиков перестановок элементов электрической схемы на монитор компьютера (рисунок 1).

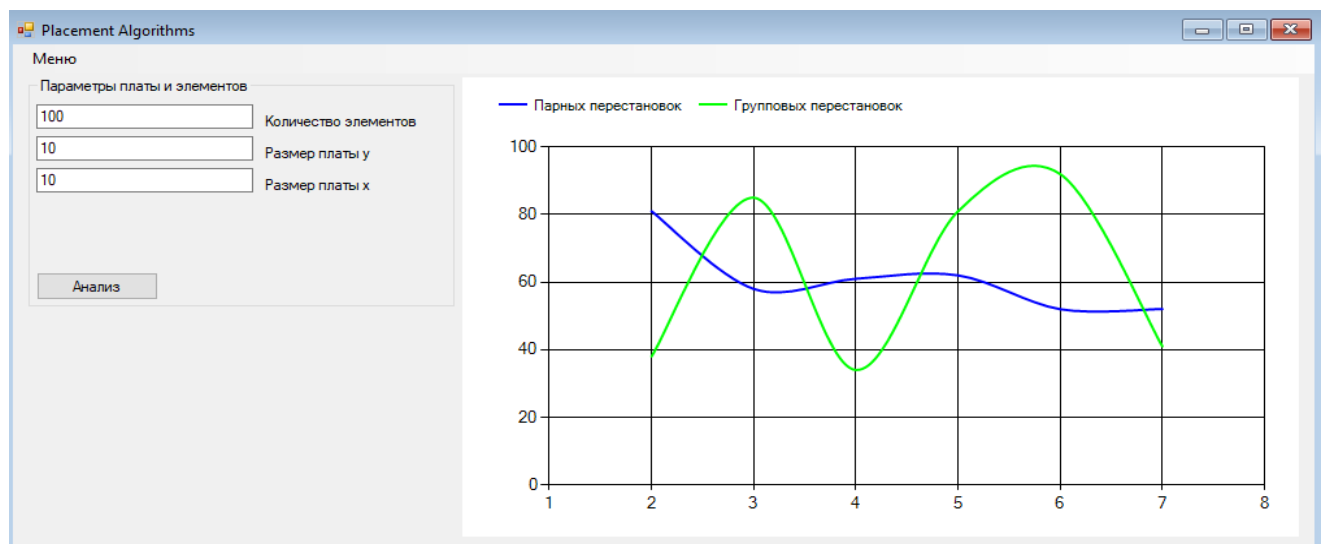


Рисунок 1 – Графики эффективности размещения элементов электрической схемы алгоритмами парных и групповых перестановок

ПС написано на языке C# [7].

Характеристики компьютера, на котором выполнена разработка:

- процессор - AMD A6-9220 RADEON R4, 5 COMPUTE CORES 2C+3G (2 CPUs), ~2.5GHz;
- память - 12288MB RAM;
- файл подкачки - 11742MB RAM;
- интегрированная видеокарта - AMD Radeon(TM) R4 Graphics;

- дискретная видеокарта - AMD Radeon (TM) 520;
- операционная система - Windows 10 Pro 64-bit (10.0, Build 19044).

При реализации алгоритмов в общем случае могут получаться локальные минимумы целевой функции. Поэтому представляют интерес получение оценок суммарной длины соединений и числа пересечений для графа электрической схемы с данными числами вершин и ребер.

Следует отметить, что применение рассматриваемых алгоритмов позволяет получать более качественные результаты по сравнению с последовательными (например, алгоритмом размещения элементов схемы по связности и матричным алгоритмом) за счет использования больших затрат компьютерного времени.

Заключение. В процессе проведения исследования разработано ПС сравнительной оценки алгоритмов парных и групповых перестановок для разного количества элементов электрической схемы при минимизации суммарной длины связей между размещаемыми элементами. Результаты работы ПС показали различные значения суммарной длины связей при реализации алгоритмов парных и групповых перестановок разного количества элементов схемы: при размещении 100 элементов применение алгоритма парных перестановок дает более равномерное изменение суммарной длины связей по сравнению с алгоритмом групповых перестановок.

Список использованных источников:

1. Морозов, К.К. Автоматизированное проектирование конструкций радиоэлектронной аппаратуры: учеб. пособие для вузов / К.К. Морозов, В.Г. Одинокое, В.М. Курейчик. – М.: Радио и связь, 1983. – 280 с.
2. Зыков, А.Г. Алгоритмы конструкторского проектирования ЭВМ / А.Г. Зыков, В.И. Поляков. – СПб: Университет ИТМО, 2014. – 136 с.
3. Омельченко, А. В. Теория графов / А.В. Омельченко. – М.: МЦНМО, 2018. – 416 с.
4. Емеличев, В. А. Лекции по теории графов. / В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. – М.: УРСС, 2009. -392 с.
5. Зыков, А.А. Основы теории графов / А.А. Зыков. – М.: Вузовская книга, 2004. - 664 с.
6. Иванова, Н.Ю. Технология проектирования печатных плат в САПР P-CAD-2006 / Н.Ю. Иванова, А.С. Петров, В.И. Поляков, Е.Б. Романова. – СПб: СПбГУ ИТМО, 2009 - 168 с.
7. Прайс, М.Д. C# 7 и .NET Core. Кроссплатформенная разработка для профессионалов / М.Д. Прайс. – СПб.: Питер, 2018. – 640 с.: ил.

UDC 621.396.6

SOFTWARE TOOL FOR ANALYZING THE EFFECTIVENESS OF ALGORITHMS FOR THE PLACEMENT OF ELECTRICAL CIRCUIT ELEMENTS

Velichko V.M., Skudnyakov Yu.A.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Skudnyakov Yu.A. – Candidate of Engineering Sciences, Associate Professor

Annotation. A software tool has been developed to perform a comparative analysis of the effectiveness of algorithms for placing electrical circuit elements on a printed circuit board with variations in their number in order to achieve a minimum total length of connections between them, which reduces the complexity, production costs of product designs, increases their reliability and durability.

Keywords. Software, efficiency, algorithms, placement, elements, electrical circuit.

УДК 621.311.25

ЗАРЯДНОЕ УСТРОЙСТВО С ИСПОЛЬЗОВАНИЕМ СОЛНЕЧНОЙ ЭНЕРГИИ

Гриневич А.И., студент

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

Шпак И.И. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В статье приведены результаты, полученные в процессе проектирования перспективного зарядного устройства. На основе обзора научно-технической литературы обоснованы современные технические требования, разработаны схемы: структурная, на её основе функциональная и далее электрическая принципиальная; рассчитаны электрические параметры функциональных узлов устройства и разработаны алгоритм и программное обеспечение его функционирования. Спроектирована также топология печатной платы и конструкция печатного узла блока управления, с применением САПР.

Ключевые слова. Солнечная электростанция, микропроцессор, радиоэлектронное устройство, инвертор постоянного напряжения в переменное, технические требования, схема структурная, схема электрическая принципиальная, широтно-импульсный модулятор, конструкторское проектирование с применением САПР.

Введение. Солнечная энергия является одним из самых перспективных среди всех энергетических ресурсов и может использоваться даже в пасмурную погоду. Скорость, с которой солнечная энергия улавливается Землей, примерно в 10 тысяч раз превышает скорость, с которой человечество потребляет энергию.

Солнечная энергия может обеспечивать тепло, охлаждение, естественное освещение, электроэнергию и топливо для множества самых разнообразных применений. Современные технологии позволяют преобразовывать солнечный свет в электрическую энергию с помощью фотоэлектрических панелей либо зеркал, концентрирующих солнечное излучение. Как показали результаты исследований NASA, в течение 6,5 миллиардов лет Солнце способно обогревать Землю своими лучами [1].

В Беларуси эффективность и производительность систем будет намного ниже, чем в солнечной Испании. И вместе с тем стоит отметить, что даже нашего солнца, с его не такой уж и большой активностью, более чем достаточно.

На рисунке 1 показано среднестатистическое распределение инсоляции (облучение поверхностей солнечным светом под разными углами наклона) на территории РБ [2].



Рисунок 1 – Среднестатистическое распределение инсоляции на территории РБ

Практически в каждом регионе Беларуси есть возможности для эффективного использования энергетического потенциала Солнца. Годовая инсоляция, т.е. облучение земной поверхности солнечной энергией, находится в пределах от 900 кВт-час/м² до 1100 кВт-час/м².

Несмотря на кажущуюся простоту устройств, преобразующих солнечную энергию в электрическую, их использование на территории РБ и РФ (средняя полоса широта Минск-Москва) очень ограничено – климатом, погодой, временем года и суток. Плюс к тому, чтобы система подавала ток в сеть, необходимо приобрести:

- аккумулятор, который будет накапливать энергию на случай перепадов напряжения;
- инвертор, который будет переводить постоянный ток в переменный;
- систему, контролирующую заряд аккумулятора.

Поэтому реализация проекта, посвященного разработке зарядного устройства с использованием солнечной энергии, является актуальной и практически значимой.

Основная часть. Для определения технических характеристик и структуры разрабатываемого устройства, будем исходить из базовой схемы соединений в системах с солнечными батареями, для случая с наличием напряжения бытовой сети, изображенной на рисунке 2.

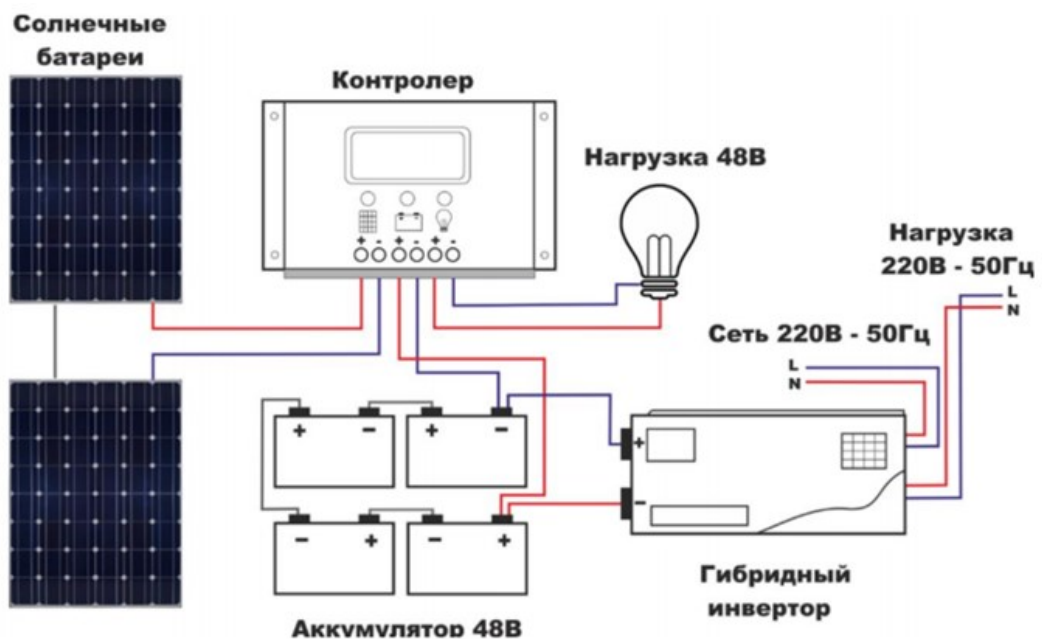


Рисунок 1 – Схематическое изображение и сама схема соединений для систем с наличием напряжения бытовой сети

Разрабатываемое устройство должно обеспечивать выполнение следующих функций:

- правильный режим зарядки аккумуляторов от солнечной панели.
- преобразование энергии получаемой от солнечной панели в напряжение бытовой сети 230 В, 50 Гц с током не менее 3 А,
- наличие разъема питания для зарядных устройств USB;
- контроль состояния аккумулятора и отключение аккумулятора от зарядки при достижении 100 % напряжения на аккумуляторе.
- возможность применения аккумуляторов 12В с возможностью определения номинала и отключением при превышении максимального напряжения;
- возможность отключения нагрузки при разряде аккумулятора и малой солнечной активности;
- индикацию состояния аккумулятора, энергетических возможностей солнечных панелей, мощности подключенной нагрузки;
- возможность коммуникации потребитель-техника;
- возможность интеграции в систему умный дом.
- встроенная защита от ошибок при подключении солнечных панелей и аккумуляторов

Для реализации данных функциональных возможностей была разработана схема на основе универсального микроконтроллера ATmega128 [3], показанная на рисунке 3.

Ядром устройства, несомненно, является блок управления на основе микроконтроллера, который выполняет функциональную роль менеджера. Основой блока управления является микроконтроллер. Выбранный тип контроллера является популярным у разработчиков радиоэлектронных устройств благодаря невысокой цене, стабильности параметров и большому количеству информационных ресурсов [4], позволяющих быстро и с небольшими финансовыми издержками освоить его.

Весьма непросто в процессе проектирования оказался выбор датчика тока. Существует множество методов измерения тока [5,6], однако в промышленности наиболее широко применяются три: на основе использования токового шунта, на основе эффекта Холла и трансформатора тока.

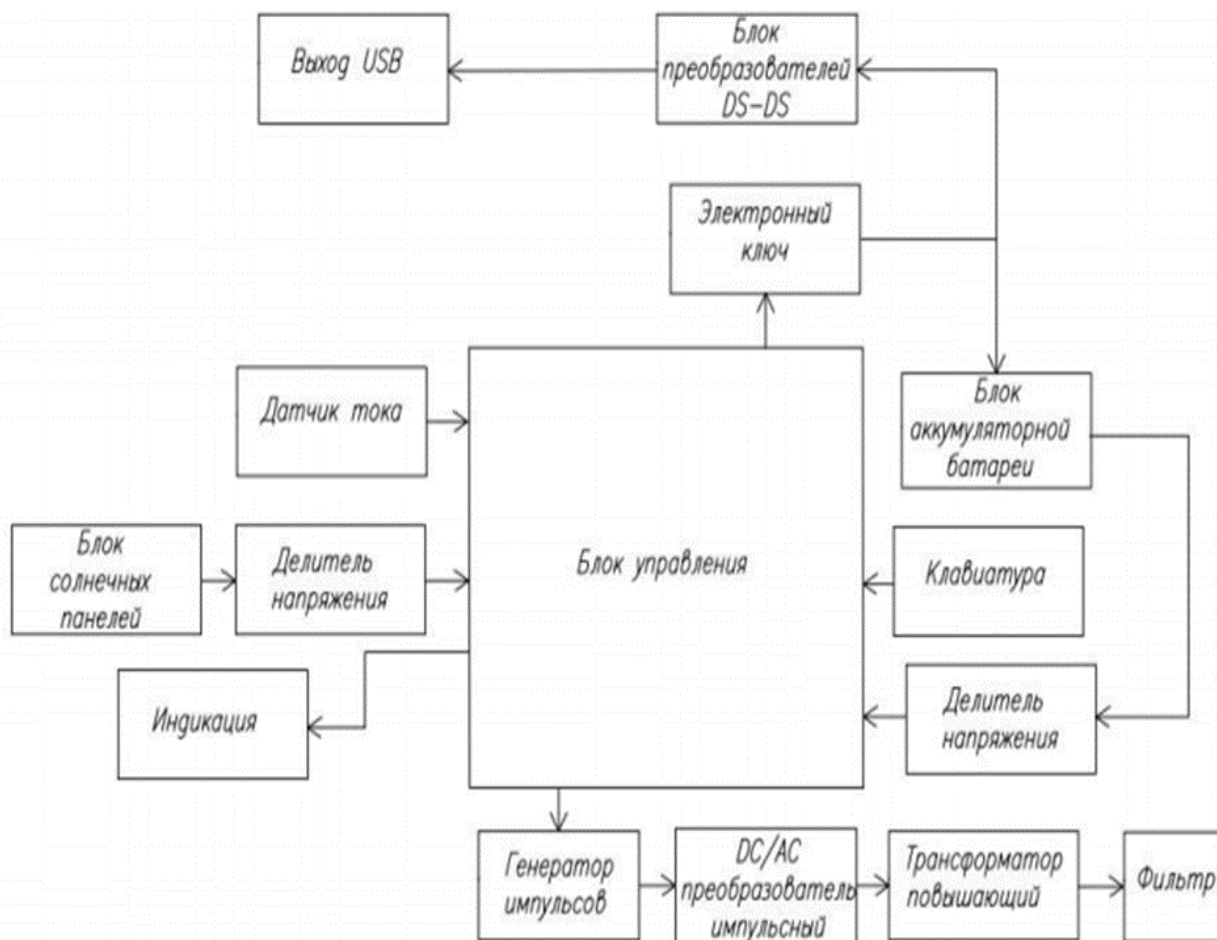


Рисунок 3 – Структурная схема зарядного устройства

Самой простой является схема измерения тока с помощью шунта. При этом измеряется падение напряжения на шунте, после чего рассчитывается значение тока. Основными недостатками здесь являются:

- отсутствие гальванической развязки;
- низкий КПД.

Указанные недостатки существенно ограничивают использование токовых шунтов.

Схемы для измерения тока на основе трансформатора тока, с помощью датчика на эффекте Холла и измерения напряжения с помощью делителя напряжения показаны на рисунке 4.

Конструкция блока управления была реализована в виде небольшого блока. В процессе проектирования топологии печатной платы и печатного узла на её основе была использована система автоматизированного проектирования Altium Designer.

Одним из преимуществ созданной системы управления микроклиматом является сравнительная простота его настройки, невысокая стоимость, а также возможность улучшения эксплуатационных характеристик и создания более сложных систем контроля и управления микроклиматом.

В результате анализа достоинств и недостатков [6] из датчиков, приведенных на рисунке, для измерения тока был выбран датчик на эффекте Холла - рисунок 4 (б).

Схема электрическая структурная и выбранные функциональные узлы, послужили основой для проектирования схемы электрической функциональной, на базе которой была разработана схема электрическая принципиальная.

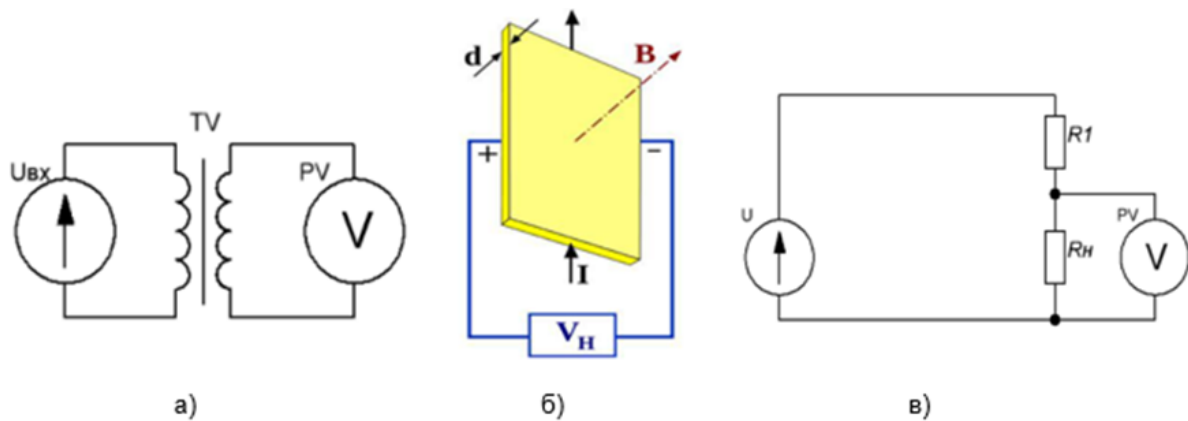


Рисунок 4 – Схемы для измерения тока и напряжения на основе трансформатора тока (а), с помощью датчика на эффекте Холла (б) и делителя напряжения (в)

Заключение. Реализовано устройство после проведённого электрического расчёта функциональных узлов с использованием современной элементной базы. Далее были разработаны алгоритм работы и программное обеспечение [4] зарядного устройства.

Конструкторское проектирование блока управления было реализовано в виде автономной конструктивной единицы. Осуществлялась разработка с использованием системы автоматизированного проектирования Altium Designer.

Весьма значимым преимуществом данной разработки является простота обслуживания, невысокая стоимость, а также возможность модернизации и создания более совершенных и многофункциональных устройств.

Список использованных источников:

1. Климат и климатические ресурсы [Электронный ресурс] – Режим доступа: <https://projecteducation.ru/explore/belarus/item/222-klimat-iklimaticheskie-resursy-belarusi> Дата доступа: 23.11.2022.
2. Карты солнечной энергии [Электронный ресурс] – Режим доступа: <https://makroterm.by/akademia-znanij/geliosistemy-statiy/karta-insolyacii-solnechnoi-radiacii-belarusi.html> Дата доступа: 23.11.2022
3. Микроконтроллеры АТmega128 [Электронный ресурс] – Режим доступа: <http://catalog.gaw.ru/> Дата доступа: 23.11.2022
4. Евстифеев, А. В. Микроконтроллеры AVR семейства Мегга: руководство пользователя / А. В. Евстифеев // - Москва: ДМК Пресс, ДОДЭКА, 2015. - 587 с.
5. Электронные компоненты. Каталог [Электронный ресурс]. – Режим доступа: <https://www.chipdip.by/> (Дата доступа: 02.04.2022).
6. Свинцов Е.Л. Современные Датчики. Справочник.: справ. пособие / Е.Л. Свинцов, Заболотной Ю.А. Москва.: Техносфера, 2005. – 588с.

UDC 621.311.25

CHARGER DEVICE ON THE BASE OF USING SOLAR ENERGY

Grinevich A.I.

Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus

Shpak I.I. – Candidate of Engineering Science, Associate Professor

Annotation. The article presents the results obtained in the process of designing a promising charger. Based on the review of scientific and technical literature, modern technical requirements are substantiated, schemes are developed: structural, functional and further electrical principle based on it; electrical parameters of functional units of the device are calculated and an algorithm and software for its functioning are developed. The topology of the printed circuit board and the design of the printed unit of the control unit are also designed, using modern CAD.

Keywords. Solar power plant, microprocessor, radio-electronic device, DC-to-AC inverter, technical requirements, structural diagram, electrical schematic diagram, pulse width modulator, design using CAD.

УДК 793.2

АВТОМАТИЗИРОВАННАЯ СИСТЕМА СЦЕНИЧЕСКОГО СОПРОВОЖДЕНИЯ ВЫСТУПЛЕНИЙ

Дунай Р.Н., студент

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Шпак И.И. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В статье приведены результаты, полученные в процессе создания автоматизированной системы сценического сопровождения выступлений. Разработанная система позволяет соответствующим образом выбрать, запрограммировать и в дальнейшем реализовывать эффективное освещение зрелищных мероприятий. Автором выполнено схемотехническое проектирование системы, разработаны алгоритм и программное обеспечение её функционирования, а также конструкция печатного узла блока управления, с применением современной САПР.

Ключевые слова. Автоматизированная система, сценическое сопровождение, осветительное оборудование, микроконтроллер, схемотехническое проектирование, разработка алгоритма и программного обеспечения, конструкторское проектирование.

Введение. Несмотря на то, что выражение «Хлеба и зрелищ!» (на латыни — «Panem et circenses!») древнеримского поэта-сатирика Деция Ювенала, жившего почти два тысячелетия назад, является стихотворным обличием Ювеналом упадка доблести и славы древнего Рима, в современном обществе этот лозунг активно подхвачен политиками и внедряется в жизнь как линия поведения через зрелищные мероприятия и средства массовой информации: кинофильмы, телевидение, компьютерные игры. Вопреки неоднозначной оценке данного лозунга в политическом контексте, он весьма успешно используется в культурном, так как способствует удовлетворению самых насущных или базовых потребностей населения, особенно во времена обострения общественно-политических и экономических кризисов.

Всё вышесказанное обуславливает исключительную актуальность разработки автоматизированной системы для сценического сопровождения зрелищных мероприятий. Создание такой целостной системы, на основе использования современного оборудования и информационно-коммуникационных технологий, позволит организаторам мероприятий и сотрудникам учреждений культуры, на базе которых мероприятия проводятся, при минимальных временных и материальных затратах обеспечить высокое качество выступлений. Одним из важнейших факторов, определяющих успех зрелищных мероприятий является соответствующим образом выбранная и реализованная система освещения. Этот выбор должен подчеркивать в наиболее выгодном свете все, что происходит на сцене любого культурного сооружения (концертного зала, сцены дома культуры, стадиона и т.д.). На сегодняшний день управление освещением является очень важной и не простой задачей [1].

Основная часть. При выборе сценического осветительного оборудования (прожекторы, софиты и прочее) следует исходить из того, что оно должно отвечать следующим требованиям:

- создавать необходимый уровень и цветовую гамму освещенности;
- быть безопасным для людей;
- иметь простое обслуживание;
- иметь высокую четкость подсветки, которая при необходимости позволяет:
- подчеркивать все действия, происходящие на сцене, чтобы они были видны со всех ракурсов и углов;
- четко подсвечивать различные элементы: декорации, исполнители, оркестр и т. д.

Разрабатываемая система сценического сопровождения, должна обеспечивать:

- перемещение при помощи пульта управления не менее трёх прожекторов, расположенных на сцене, по горизонтали (ось X), перемещение должно осуществляться для каждого прожектора отдельно;
- перемещение при помощи пульта управления не менее трёх прожекторов, расположенных на сцене, по вертикали (ось Y), перемещение должно осуществляться для каждого прожектора отдельно;
- сохранение данных координат положений прожекторов, индивидуально для любого из трёх прожекторов;
- автоматическое перемещение любого из трёх прожекторов, по ранее сохранённым данным координат их положений;
- переключение мощности освещения каждого из трёх прожекторов, переключение цветов прожекторов; переключение должно осуществляться для каждого прожектора индивидуально.

Схема электрическая структурная системы сценического сопровождения выступлений, обеспечивающей реализацию перечисленных функций, изображена на рисунке 1. Цепи питания прожекторов на схеме не отображены.

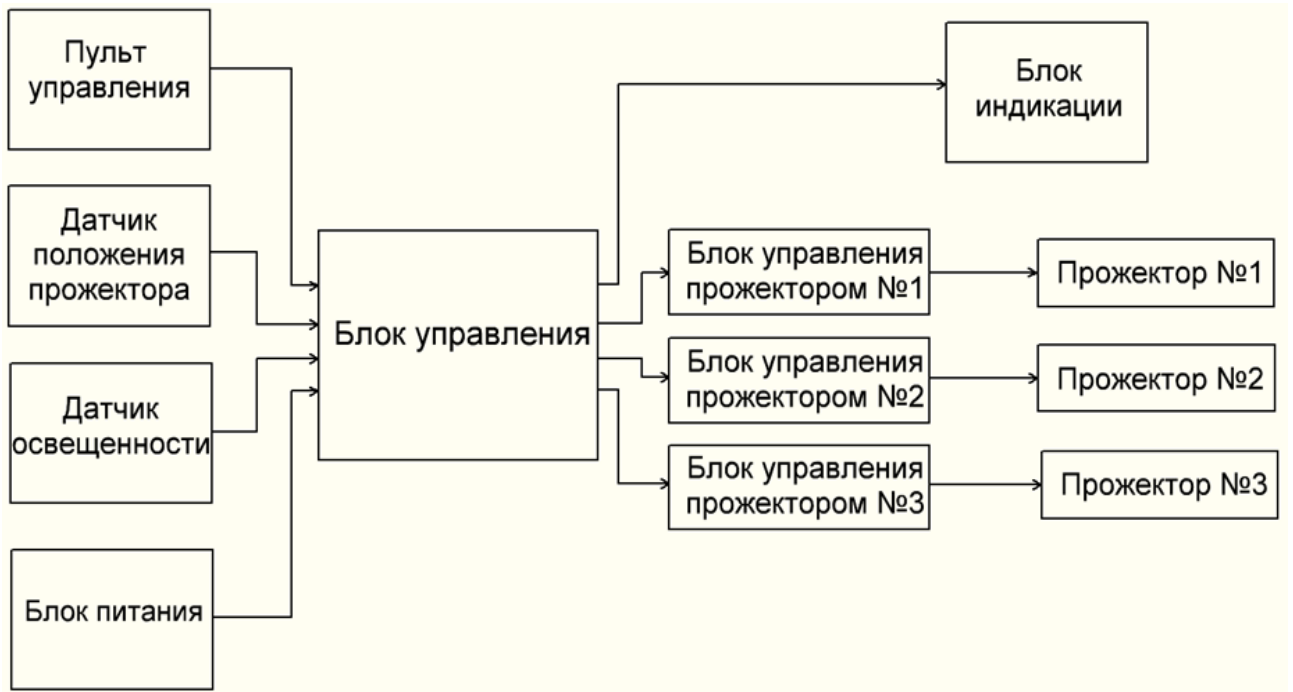


Рисунок 1 – Схема электрическая структурная автоматизированной системы сценического сопровождения выступлений

Система начинает свою работу после подачи напряжения с блока питания. Блок управления анализирует состояние кнопок пульта управления, после которого получает информацию о том, какой из прожекторов должен быть включён—выключен, и как каждый из прожекторов должен освещать сцену, а также какой из трёх прожекторов необходимо повернуть по оси X или оси Y.

Далее блоки управления каждого из трёх прожекторов формируют сигналы для блока индикации, для того чтобы информировать оператора о том, какие прожекторы в данный момент работают, и что система работает без сбоев.

Если нажата кнопка перемещения одного из прожекторов по оси X или Y, то блок управления перемещением прожектора формирует управляющий сигнал, на соответствующее поворотное устройство нужного прожектора.

При нажатии на пульте управления кнопки запоминания текущего положения прожектора, в память блока управления перемещением прожектора записывается информация координат нахождения прожектора в данный момент времени.

При нажатии на пульте управления кнопки автоматического перемещения прожектора, на поворотное устройство соответствующего прожектора поступит управляющий сигнал с блока, и прожектор поменяет своё положение на координаты, ранее сохранённые в память блока управления перемещением прожекторов.

После коммутации соответствующих кнопок на пульте управления, касающихся включения и выключения прожекторов, или изменения силы свечения прожекторов, блок управления освещением прожекторов формирует соответствующие сигналы на прожектора. Если один из прожекторов в данный момент времени включен, то на блоке индикации будет светиться светодиод (для каждого прожектора свой). Если прожектор в данный момент времени выключен, то соответствующий диод не будет светиться.

Датчик освещения в системе используется для автоматического управления источниками искусственного (электрического) света. В зависимости от степени освещения окружающего пространства датчик способен подавать сигнал для включения/выключения ламп, прожекторов, фонарей и других осветительных приборов. Датчик позволяет отслеживать и контролировать яркость освещения определенной территории или помещения.

Ядром системы, несомненно, является блок управления. Он должен считывать информацию с пульта управления, принимать решения по включению и выключению прожекторов. Поэтому его следует реализовать на основе микроконтроллера.

Анализ требований белорусских проектировщиков показывает, что наиболее востребованными на белорусском рынке являются контроллеры с архитектурами PIC и AVR.

Микроконтроллеры PIC появились раньше микроконтроллеров AVR и в настоящее время имеют широкую номенклатуру, а также хорошо освещены в литературе. К недостаткам данных контроллеров следует отнести:

- выполнение операций за 4 такта (против 1 такта для большинства операций в AVR), что требует повышения рабочей частоты микроконтроллера;
- выполнение большинства операций через регистр W;
- менее развитую, по сравнению с AVR, систему команд;
- страничную организацию памяти;
- небольшой размер стека;
- большую цену.

Поэтому в разрабатываемой системе будет применён микроконтроллер с архитектурой AVR. В настоящее время существует два основных семейства микроконтроллеров с архитектурой AVR: Tiny и Mega. Семейства Mega обладают большими возможностями по сравнению с микроконтроллерами семейства Tiny: большее число портов ввода/вывода, больше объемы памяти программ и данных, улучшенная система команд. Поэтому следует применять микроконтроллер из семейства Mega [2]: ATmega168, ATmega48 т.д.

ATmega48/ATmega88/ATmega168 — низкопотребляющие 8 битные КМОП микроконтроллеры с AVR RISC архитектурой. Выполняя команды за один цикл, ATmega48/ATmega88/ATmega168 достигают производительности 1 MIPS при частоте задающего генератора 1 МГц, что позволяет разработчику оптимизировать отношение потребления к производительности.

Общий блок управления и блоки управления каждым прожектором реализованы на основе микроконтроллера. Был выбран микроконтроллер серии ATmega [3], архитектура которого представлена на рисунке 2.

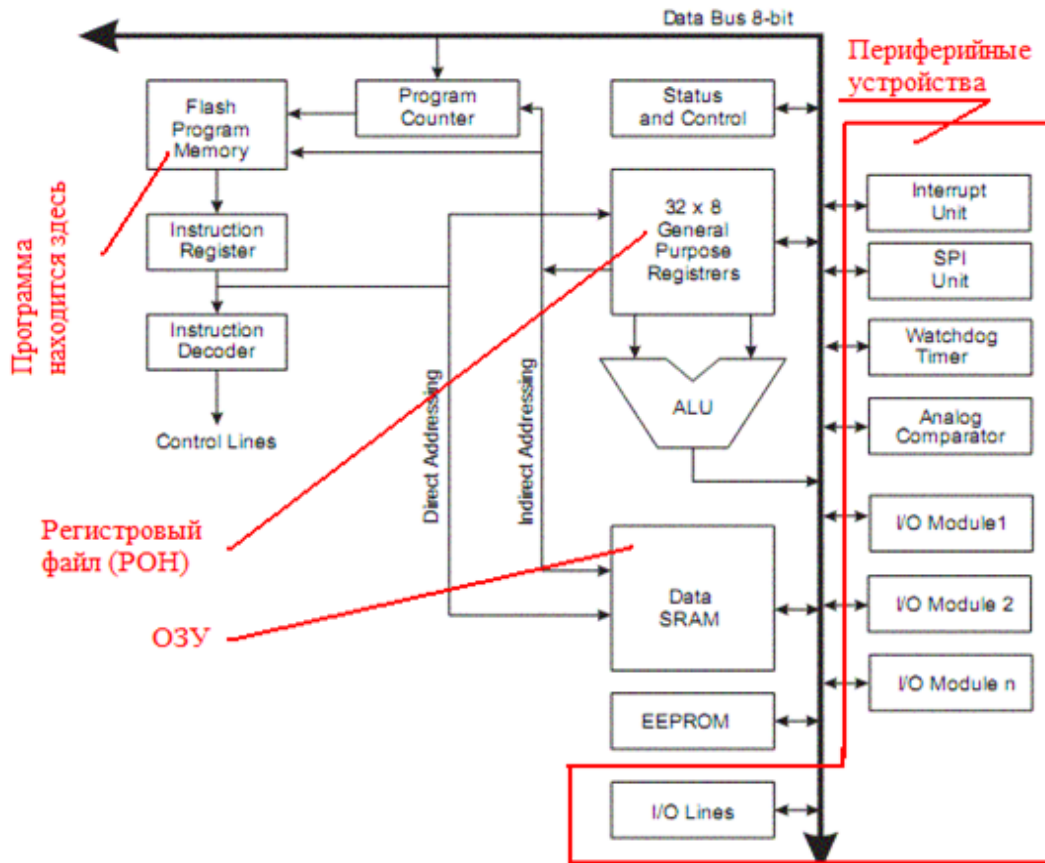


Рисунок 2 – Архитектура микроконтроллера ATmega

Данный тип контроллера широко используется в мире учитывая невысокую цену, стабильность параметров и обилие информационных ресурсов [4], позволяющих изучить его в кратчайшие сроки и с небольшими финансовыми издержками.

Исполнительными устройствами для общего блока управления являются три блока управления прожекторами. Микроконтроллер обеспечивает переключение цвета прожектора, а также силу свечения. В случае необходимости отключения прожектора, предусмотрена функция отключения для каждого прожектора.

Чтобы обеспечить требования по перемещению прожекторов по осям X и Y, целесообразно применение поворотных устройств на основе шаговых двигателей. Прожектор освещения должен быть жёстко закреплён на валу шагового двигателя, тот в свою очередь закреплён на валу второго

шагового двигателя, последний прикреплён к кронштейну на потолке или стене. Связка двухшаговых двигателей позволяет перемещать прожектор в плоскостях (X) и (Y) с углами по 360 градусов. Управление первым шаговым двигателем необходимо для поворота прожектора по горизонтали (ось X), управление вторым шаговым двигателем осуществляет поворот прожектора по вертикали (ось Y).

Для управления поворотом прожекторов выбраны шаговые двигатели типа 28BYJ-48. Преимущества данного шагового двигателя в том, что у него не большие габаритные размеры, малая стоимость, доступность. Соотношение 64:1 создает на валу шагового двигателя достаточное усилие, чтобы производить управление — движение прожекторами. Драйвер управления униполярным шаговым двигателем 28BYJ-48, по сравнению с аналогичными, схематически прост.

На основе схемы электрической функциональной структурной и выбранных функциональных узлов, спроектирована схема электрическая функциональная автоматизированной системы сценического сопровождения, представленная на рисунке 3, которая послужила в дальнейшем основанием для разработки схемы электрической принципиальной.

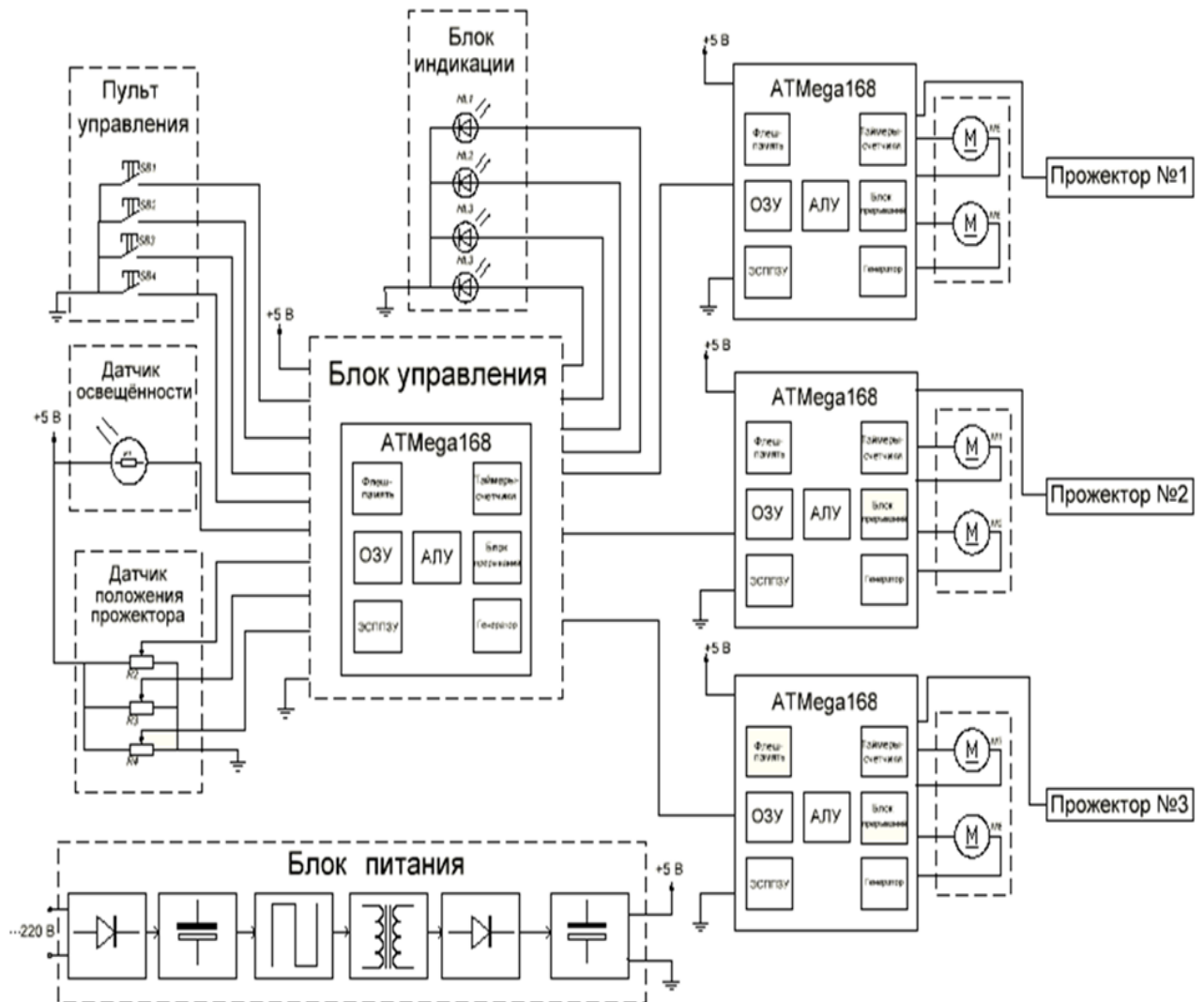


Рисунок 3 – Схема электрическая функциональная автоматизированной системы сценического сопровождения выступлений

В программном обеспечении автоматизированной системы сценического сопровождения предусмотрена настройка микроконтроллера для выработки управляющих сигналов на шаговые двигатели и прожектора. Чтобы считывать данные с датчиков и кнопок, в программном обеспечении предусмотрено считывание данных с портов ввода-вывода микроконтроллера. Хранение координат, по которым в автоматическом режиме происходит перемещение прожекторов, необходимо реализовать в энергонезависимой памяти микроконтроллера.

При выборе датчиков ориентировались в первую очередь на использование широко распространённых в настоящее время множество датчиков со встроенным микропроцессором. Обработка измерений непосредственно в самом датчике позволяет улучшить его характеристики. Общая схема таких датчиков [5], показана на рисунке 4.

Сам датчик находится под управлением микропроцессора, это позволяет менять уровни возбуждения так, что переключение диапазонов происходит автоматически. Кроме того, могут

отслеживаться температура окружающей среды и другие параметры, а также производится коррекция показаний датчика с использованием аналогового усилителя или данных, сохраненных в памяти.



Рисунок 3 – Архитектура датчиков со встроенным микропроцессором

Заключение. Наиболее ценная возможность, предоставляемая микропроцессорной обработкой – это возможность получения линейного сигнала от нелинейного датчика при помощи таблицы соответствия, хранящейся в памяти. Это позволяет создавать датчики с очень низкой нелинейностью.

Список использованных источников:

1. Варианты освещения сцены: какой должна быть подсветка, схемы [Электронный ресурс]. – Режим доступа: <https://1posvetu.ru/svetodizajn/osveshhenie-stseny.html>. Дата доступа: 23.11.2022.
2. ATmega48, ATmega88, ATmega168 [Электронный ресурс]. – Режим доступа: http://www.gaw.ru/html/cgi/txt/ic/Atmel/micros/avr/atmega48_88_168.htm. Дата доступа: 23.11.2022
3. Евстифеев, А. В. Микроконтроллеры AVR семейства Mega: руководство пользователя / А. В. Евстифеев // - Москва: ДМК Пресс, ДОДЭКА, 2015. - 587 с.
4. Языки программирования микроконтроллеров [Электронный ресурс]. – Режим доступа: <https://mcscpu.ru/index.php/soft/42-Imcu/67-programmlang>. Дата доступа: 02.04.2022.
5. Джексон, Р. Г. Новейшие датчики / Р.Г. Джексон// – Москва: Техно-сфера, 2000. – 256 с.

UDC 793.2

AUTOMATED SYSTEM OF STAGE ACCOMPANIMENT OF PERFORMANCES

Dunaj R.N.

Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus

Shpak I.I. – Candidate of Engineering Science, Associate Professor

Annotation. The article presents the results obtained in the process of creating an automated system for stage accompaniment of performances. The developed system allows you to select, to program, and to implement the effective lighting of entertainment events accordingly. The author of the article performed circuit design of the automated system, developed an algorithm and software for its operation, as well as the design of the printed unit of the control unit, using modern CAD.

Keywords. Automated system, stage accompaniment, lighting equipment, microcontroller, circuit design, algorithm and software development, design engineering.

УДК 004.42

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ ВЕРСТКИ И ПОДГОТОВКИ К ПЕЧАТИ МАТЕРИАЛОВ ДЛЯ НАСТОЛЬНЫХ ИГР

Желенок Д.А., студент, Савенко А.Г., маг. техн. наук

*Белорусский государственный университета информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Савенко А.Г. маг. техн. наук, ст. препод. каф. ИСиТ

Аннотация. В работе представлено программное средство, предназначенное для автоматизации верстки и подготовки к печати материалов настольных игр. Программное средство имеет графический и консольный интерфейс, продуманную архитектуру, позволяющую не дублировать исходный код алгоритмов для различных интерфейсов. Основными функциями программного средства являются: подготовка файла спецификации с широким выбором параметров (различные виды материалов полиграфической продукции, форматы листов (в том числе рулонная печать), варианты исходных файлов материалов настольных игр и способов их сопоставления, выбор стратегии упаковки); генерация документов для печати в соответствии со спецификацией. Алгоритмы программного средства решают задачи двумерной упаковки и гильотинного раскроя. Применение программного средства позволит сократить затраты на подготовку и производство настольных игр.

Ключевые слова. Издание полиграфической продукции, бизнес-процесс, оптимизационные задачи, задача двумерной упаковки, задача гильотинного раскроя, автоматизация процесса, файл спецификации.

Введение. Сегодня настольные игры являются распространенным видом увлечения, образующим целую индустрию. Они имеют большое разнообразие как в оформлении и тематике, так и в жанрах, начиная от абстрактных и логических игр, заканчивая стратегическими и ролевыми играми. Существует много сообществ, клубов, а также интернет-ресурсов, посвященных настольным играм.

Процесс разработки новых игр или дополнений к существующим, чаще всего является итеративным. Это означает что, прежде чем создать финальную версию продукта, разрабатывается прототип, на нем проверяются заложенные идеи и механики через непосредственно, игру, затем вносятся правки, снова проверяются заложенные идеи и т.д. Каждый цикл прототипирования и проверки является отдельной итерацией разработки игры. После разработки готовой версии продукта автор может связаться с издателями, чтобы представить им свои наработки и, возможно, издать через них готовую к продаже игру. Также он может попробовать самостоятельно издать ее, работая напрямую с типографиями. Или как альтернативный вариант, выпустить ее материалы в цифровом виде, как бесплатно, так и платно через специализированные ресурсы.

Кроме того, очередное переиздание тиражей настольных игр нередко сопровождается внесением изменений в правила и механики игры. Изменения в правилах также не редко сопровождаются изменением как информации, содержащейся на игровых компонентах, так и самих компонентов.

Основная часть. Итеративный подход и выпуск или перевыпуск настольной игры в любом виде, подразумевает выполнение верстки компонентов, для дальнейшей их печати. Упрощение и увеличение качества верстки позволяет уменьшать расходы на издание настольных игр. Это, в свою очередь, позволяет снизить стоимость отдельных экземпляров, что повышает доступность, позволяющую охватить большую аудиторию.

Для изготовления различных видов компонентов используются различные материалы. Так, например, для игровых костей чаще всего используется пластмасса, для планшетов и полей плотный картон, а для карт картон или плотная многослойная бумага. Таким образом разные компоненты требуют разной процедуры подготовки для дальнейшего изготовления.

Подготовка компонентов к изготовлению отличается на этапе разработки и издания. Так из-за итеративности процесса разработки, изменения вносятся часто, но направлены они преимущественно на правила и игровые механики, а не на дизайн компонентов (хотя, и он может продумываться и изменяться на данном этапе). На этапе издания дизайн должен приобрести свой конечный вид, который будет удобен как в производстве, так и конечному потребителю.

Для таких компонентов как книги правил, планшеты, поля и карты необходимо предварительно сверстать документ, который можно будет распечатать. Версткой документа называется расположение текста, таблиц, картинок и других элементов на странице документа. В контексте настольных игр процесс верстки подразумевает расположение игровых элементов, так чтобы их было удобнее напечатать и разрезать. При этом формат бумаги и компоновка объектов зависит от того, где в дальнейшем изготавливаются сверстанные объекты. Так на этапе разработки игры логично форматировать карты на листах А4, это позволит легко распечатать их на домашнем принтере или в местном центре полиграфии. Для типографий может больше подойти широкоформатная или рулонная печать.

Разработанное программное средство (ПС) позволяет автоматизировать процесс верстки и подготовки материалов настольных игр к печати. ПС поддерживает гибридный режим работы, что означает поддержку как графического, так и текстового (консольного) варианта интерфейса. Графический режим работы предоставляет пользователю удобный интерфейс для создания файла спецификации и его применение для генерации документов. Файл спецификации – это файл в формате YAML который в декларативной форме описывает параметры, которые ожидаются от сформированных документов. YAML – это удобный для пользователя язык сериализации данных для всех языков программирования. Расшифровка YAML является рекурсивным акронимом и расшифровывается как «YAML Ain't Markup Language» («YAML - не язык разметки») [1]. Использование данного формата предоставляет как простую обработку со стороны программы, так и возможность вручную отредактировать файл без необходимости использовать графический интерфейс.

Файл спецификации описывает следующие параметры:

- структуру папок, в которые будут сохранены результат работы программы;
- количество и название документов, которые пользователь ожидает получить;
- формат бумаги и отступы, которые должны быть соблюдены в каждом из документов;
- источник изображений игровых компонентов и их ожидаемый размер;
- стратегия упаковки объектов на каждой странице;
- источник изображений для обложек, если компоненту они требуются;
- способ формирования меток для реза, если они необходимы.

Каждый из пунктов приведенных выше позволяет настроить ожидаемый результат. Программа, считывая спецификацию и остальные необходимые файлы, может предоставить следующие функции:

- расстановка изображений на двумерной ограниченной области с учетом отступов, ожидаемых размеров компонентов и выбранной пользователем стратегии упаковки;
- масштабирование входных изображений, а также формирование рамки вокруг них в том случае, если необходимо избежать проблем, связанных со смещением бумаги во время печати;
- нарезка входных изображений, в случае если они не влезают в ожидаемую область;
- расстановка меток для реза;
- генерация и расстановка рубашек для тех компонентов, которым она необходима;
- возможность распределить компоненты как в рамках одного документа, так и в разных.

На выходе программа формирует структуру из папок, в которых находятся сгенерированные файлы в формате Portable Document Format (PDF). Структура папок также описывается в исходной спецификации.

Упаковка – класс задач комбинаторной геометрии, включающих в себя размещение фигур заданного размера или формы внутри другой заданной фигуры с наибольшей экономией или с некоторыми другими ограничениями [2]. Под стратегией упаковки подразумевается алгоритм, по которому будет осуществляться расстановка игровых компонентов на страницах документов. В разработанном программном средстве алгоритмы упаковки подразделяются на две основные группы:

- оптимизирующие используемое пространство;
- упрощающие послепечатную обработку.

Первые представляют из себя алгоритмы, решающие задачу двумерной упаковки (2-Dimensional Bin Packing, 2DBP). Они должны быть способны работать с компонентами разных размеров и их результат должен, по возможности, минимизировать занимаемое на листах пространство и количество используемых листов бумаги.

Вторые должны минимизировать количество гильотинных разрезов, которые необходимо совершить чтобы вырезать все компоненты. Эта проблема также известна как задача гильотинного раскроя (Guillotine cutting).

Задача, решаемая алгоритмами двумерной упаковки, формулируется следующим образом: даны прямоугольники определенного размера (задана ширина, высота), которые необходимо расположить на прямоугольниках большего размера (контейнеры). Основной целью является нахождение такого расположения всех прямоугольников, чтобы они не пересекались и занимали наименьшее количество контейнеров [3]. У задачи двумерной упаковки есть два варианта входных данных: когда набор упаковываемых объектов известен заранее (offline-проблема) и когда данные поступают порциями (online-проблема). В разработанном программном средстве применяются алгоритмы, решающие offline-проблему, так как размер всех игровых компонентов известен заранее. Еще одной вариацией задачи двумерной упаковки, которая алгоритмически реализована в данном программном средстве, является задача двумерной упаковки в двумерную полуограниченную полосу (2-Dimensional Strip Packing, 2DSP). Данный вариант отличается от 2DBP тем что упаковка осуществляется только в один контейнер, который ограничен по ширине, но не ограничен по высоте. Вместо минимизации количества контейнеров, мы минимизируем высоту заполненности одного [4].

Задачи 2DBP и 2DSP относятся к классу NP-трудных задач [3, 4]. Из-за этого исследования сфокусированы главным образом на разработке приближенных алгоритмов решения. Приближенные алгоритмы находят оптимальное решение с определенной точностью, но не гарантируют

оптимальной упаковки для любого набора данных. При этом критерий оптимальности зависит от того, что должно быть оптимизировано.

Задача гильотинного раскроя – задача комбинаторной геометрии, близкая к задаче раскроя и задачам упаковки в контейнеры. Основной целью является получение максимального числа листов прямоугольного размера из листа большего размера, делая только гильотинные разрезы, то есть прямые разрезы от края до края. Задача гильотинного раскроя также относится к классу NP-трудных задач [5].

Диаграмма вариантов использования разработанного программного средства представлена на рисунке 1.

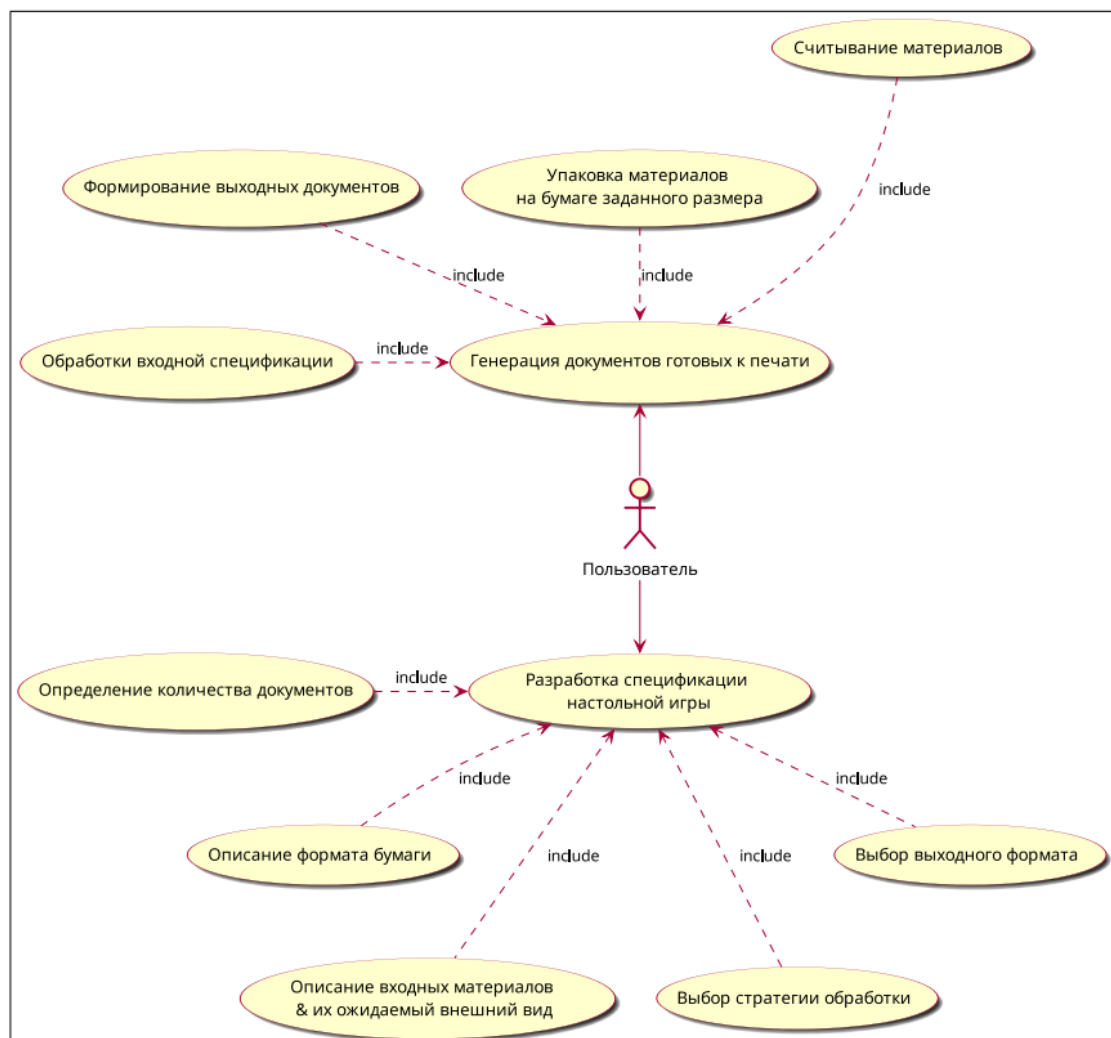


Рисунок 1 – Диаграмма вариантов использования программного средства

Из диаграммы вариантов использования (рисунок 1) видно, что все функции программного средства условно можно подразделить на две категории: одна позволяет составить спецификацию, описывающую как подготовить исходные данные и преобразовать их в документ готовый к печати, вторая позволяет совершить само преобразование входных материалов в итоговый документ.

Разберем подробнее процесс преобразования и генерации документа. Для дальнейшего моделирования понадобятся такие примитивы как точка, размер и отступы. Они в свою очередь объединяются в более комплексные сущности: обычную бумагу, рулонную бумагу и компоненты. Упаковка объектов в двумерном ограниченном пространстве осуществляется стратегиями упаковки. Для того чтобы структурировать модель, применяется шаблон проектирования «стратегия». Стратегия – это поведенческий шаблон проектирования, он определяет семейство алгоритмов, инкапсулирует каждый из них и делает их взаимозаменяемыми. Стратегия позволяет изменять алгоритмы независимо от клиентов, которые ими пользуются [6]. Стратегии упаковки возвращают упакованный документ. В нем описываются какие примитивы, где должны находиться. Такой подход позволяет не привязываться к конкретному формату для экспорта, реализовав его отдельно.

Бумага, набор компонентов и стратегия упаковки позволяют представить спецификацию документа. После создания упакованного документа необходимо преобразовать его в файл или вывести на экран. Для этого еще раз используется шаблон проектирования «стратегия», но на этот раз с его помощью описываются классы для различных способов отрисовки.

Общий процесс обработки, начиная с разбора спецификации и заканчивая выходными документами можно представить в виде диаграммы состояний (рисунок 2). На ней видно, что после считывания и обработки спецификации, подготовку каждого отдельного документа можно выполнить независимо от выполнения других.

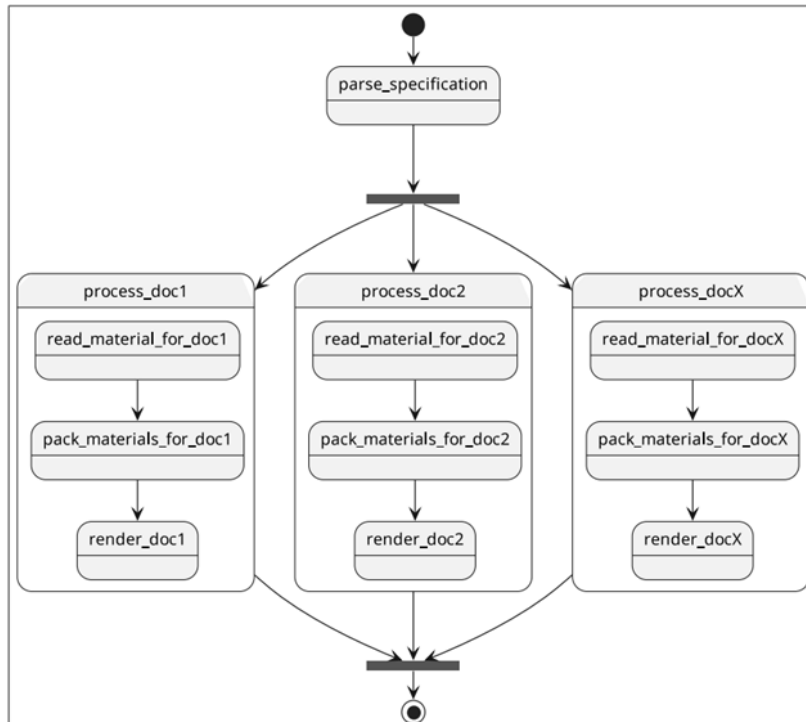


Рисунок 2 – Диаграмма состояний для генерации документа

Архитектура разработанного программного средства представляет собой три основных модуля:

- core библиотека, содержащая основные классы и функции приложения, без привязки к конкретному способу отображения;
- cli – реализация консольного интерфейса с использованием пакета click;
- gui – реализация графического интерфейса с использованием фреймворка kivy.

Консольный и графический интерфейс не зависят друг от друга, но опираются на функции и классы, предоставляемые библиотекой core. Такое разделение позволяет снизить связанность кода и уменьшает дублирование кода, позволяя реализовывать только необходимый функционал.

Библиотека может представлять собой API программного средства. Это означает что ее можно встроить в другие приложения, без необходимости устанавливать код, связанный с реализацией консольного или графического интерфейса.

Core состоит из следующих подмодулей:

- spec реализация парсера для спецификации, а также классов позволяющих манипулировать ей в объектно-ориентированном виде;
- binpack реализация различных стратегий упаковки;
- render реализация классов связанных с итоговой обработкой упакованных документов;
- pipeline реализация классов позволяющих объединить логику, представляемую подмодулями spec, binpack и render;
- measures реализация парсера и классов связанных с обработкой размеров;
- utils набор вспомогательных функций.

Обработка всех или некоторого подмножества документов, перечисленных в файле спецификации, осуществляется с помощью класса BuildPipeline. Ему на вход поступает частично обработанные данные спецификации, на основании которых осуществляется подготовка таких параметров как: формат бумаги; стратегия упаковки и рендерер.

Затем проверяет совместимость между ними, чтобы предотвратить проблемы, которые могли бы появиться на последующих этапах. Если проблем не обнаружилось, осуществляется упаковка документа на основе заданного алгоритма, а также его рендер. Обработанный документ сохраняется в виде файла или выводится на экран в зависимости от типа рендерера который использовался на предыдущем этапе.

Схема работы алгоритма конвейерной обработки документа представлена на рисунке 3а.

Перед упаковкой компонентов выполняется подготовка. Она включает в себя считывание и преобразование входных изображений фронтальной и обратной части компонентов. Схема работы алгоритма представлена на рисунке 3б.

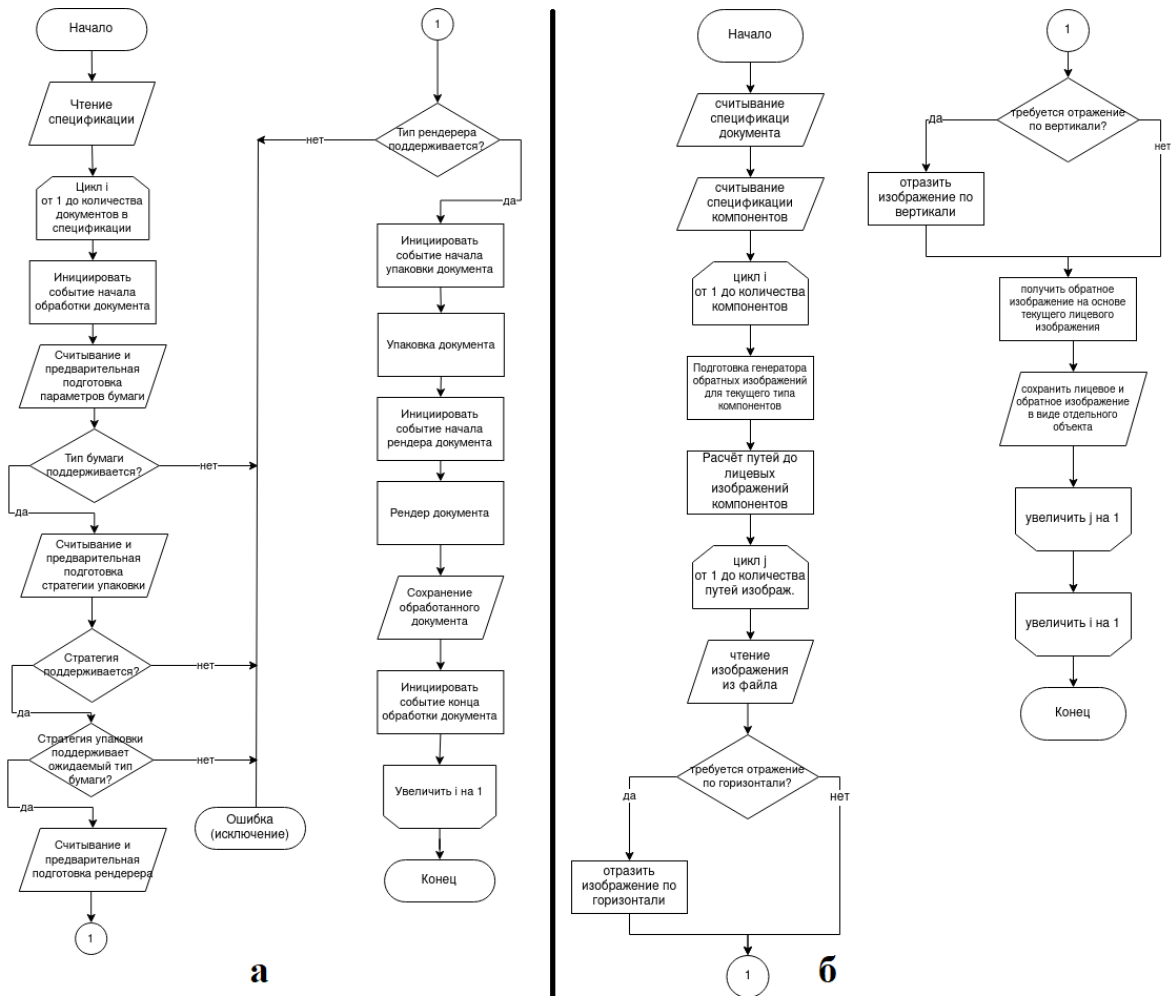


Рисунок 3 – Блок-схема алгоритмов

Во время конвейерной обработки документов необходимо находить пути к изображениям, которые должны быть вставлены в итоговый файл. Пути к ним задаются в виде двух частей, префиксной и постфиксной. Префиксы описываются в свойствах отдельного документа и представляют из себя список шаблонов путей, в которых должен начинаться поиск. Постфиксные пути же задаются в свойствах отдельного компонента. Алгоритм объединения префиксов и постфиксов представлен на рисунке 4.

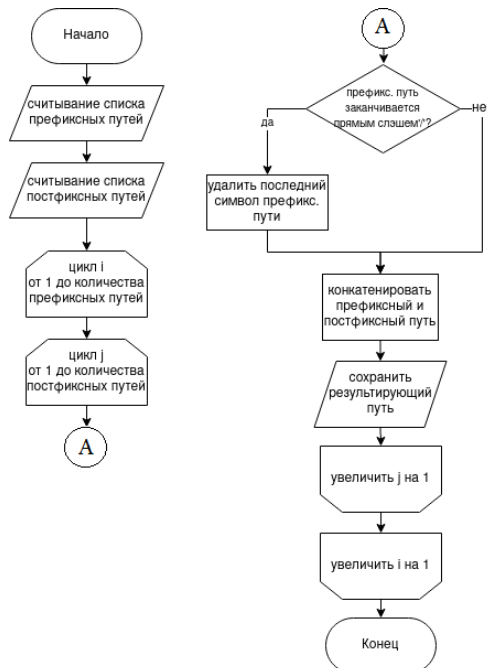


Рисунок 4 – Блок-схема алгоритма объединения префиксных и постфиксных путей

Пример работы сборки документа с предпросмотром страницы представлен на рисунке 5.

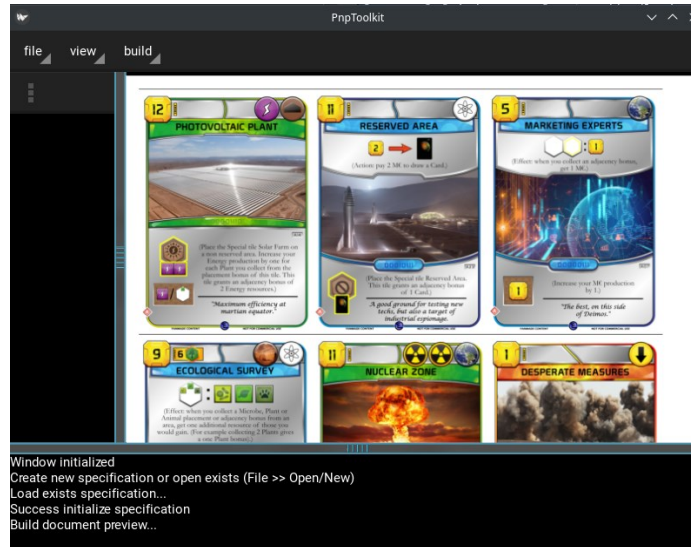


Рисунок 5 Пример сборки документа с предпросмотром страницы

Заключение. Таким образом, было разработано программное средство для автоматизации верстки и подготовки к печати материалов для настольных игр, решающее задачи оптимальной упаковки в двумерном пространстве и последующей задачи гильотинного раскроя. Разработанное программное средство может быть использовано как на производстве издателями настольных игр, так и частными пользователями для подготовки и печати компонентов настольных игр способом PNP (print and play) или разработке собственных игр. Использование разработанного программного средства позволит сократить затраты на подготовку и издание или переиздание настольных игр и существенно упростить процесс подготовки и производства.

Список использованных источников:

1. The Official YAML Web Site [Электронный ресурс]. — Режим доступа: <https://yaml.org>.
2. Britannica Packing Definition [Электронный ресурс]. — Режим доступа: <https://www.britannica.com/science/packing-combinatorics>.
3. Nikhil Bansal, Maxim Sviridenko. *New Approximability and Inapproximability Results for 2-Dimensional Bin Packing* / Maxim Sviridenko Nikhil Bansal. — 2003.
4. Habr: про двумерную упаковку: offline алгоритмы [Электронный ресурс]. — Режим доступа: <https://habr.com/en/post/136225>.
5. Arindam Khan, Arnab Maiti. *On Guillotine Separable Packings for the Two-dimensional Geometric Knapsack Problem* / Arnab Maiti Arindam Khan. — 2021.
6. Э. Гамма, Р. Хелм. *Приёмы объектно-ориентированного проектирования паттерны проектирования* / Р. Хелм Э. Гамма. — Питер, 2017. — 368 с.

UDC 004.42

SOFTWARE TOOL FOR AUTOMATION OF LAYOUT AND PREPARATION FOR PRINTING OF MATERIALS FOR BOARD GAMES

Zhelenok D.A., Savenko A.G.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Savenko A.G. – Master of Engineering Sciences

Annotation. The paper presents a software tool designed to automate the layout and preparation for printing board game materials. The software tool has a graphical and console interface, a well-thought-out architecture that allows you not to duplicate the source code of algorithms for different interfaces. The main functions of the software are: preparation of a specification file with a wide range of parameters (various types of printed materials, sheet formats (including roll printing), options for source files of board game materials and methods for their comparison, selection of a packaging strategy); generation of documents for printing in accordance with the specification. Algorithms of the software tool solve the problems of two-dimensional packing and guillotine cutting. The use of the software tool will reduce the cost of preparation and production of board games.

Keywords. Publication of printed products, business process, optimization tasks, two-dimensional packaging problem, guillotine cutting problem, process automation, specification file.

УДК 004.42

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ МЕДИЦИНСКОГО ЦЕНТРА

Колдунова А.А., студент, Ладычук И.С., студент

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Бакунов А.М. – маг. техн. наук, ст. препод. каф. ИСиТ

Аннотация. В статье рассмотрены вопросы разработки программного средства по автоматизации работы медицинского центра. Проведен сравнительный анализ существующих решений для различных медицинских учреждений, выявлены их преимущества и недостатки, приведены примеры элементов интерфейса разрабатываемого программного средства.

Ключевые слова. Программное обеспечение, разработка приложений, медицина, медицинское обслуживание.

Введение. Как известно различные информационные технологии плотнее и плотнее вплетаются в жизнь человека. Подобный прогресс уже автоматизировал полностью или частично множество процессов. Таким образом, стали менее трудозатратны различные задачи, от самых тривиальных, до объемных и сложных.

Все крупные бизнес-процессы уже прошли процесс автоматизации или вступают в него. Естественно, это связано с финансовыми возможностями организации, которой необходимо внедрение современных информационных технологий. Чаще всего финансовые возможности на внедрения различных решений имеют коммерческие и/или частные организации. Государственные предприятия не всегда имеют возможность быстрого внедрения решений для автоматизации, это связано как с финансированием, так и с более усложненными процедурами.

Существует огромное количество государственных предприятий и организаций, которым актуально пройти автоматизацию некоторых процессов, для повышения качества их работы. Одним из таких являются учреждения здравоохранения.

Медицинские услуги всегда были и будут востребованы среди населения. Важным бизнес-процессом является взаимодействие учреждения здравоохранения и пациента. Поэтому в данный процесс требуется внедрение разнообразных программных средств для отображения актуальной информации об учреждении, всей необходимой информации для пациентов, врачей и так далее.

Основная часть. В ходе первой части работы были проанализированы существующие решения для медицинских центров, произведено проектирование базы данных, произведена разработка некоторых модулей веб-приложения. Апробация материалы прошли на Международной научно-практической конференции [1].

Данная описывает такие этапы проведения автоматизации как проектирование и разработку дополнительных модулей для расширения функционала проекта.

Основные недостатки и преимущества уже существующих решений:

- загруженность пользовательского интерфейса;
- устаревший интерфейс, некачественная графика.

В качестве преимуществ выделим:

- наличие актуальной информации;
- возможность записи на прием в удобное для пациента время;
- адаптивный интерфейс.

Таким образом, дальнейшая разработка должна учитывать существующие недостатки конкурентов и преимущества.

Важным этапом жизненного цикла любого программного продукта является этап проектирования. На данном этапе определяется то, как должно работать разрабатываемое приложение.

Необходимо определить то, как и кем будет использоваться данное веб-приложение. Данную информацию лучше всего отображает диаграмма вариантов использования.

Диаграмма вариантов использования – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей [2].

Отобразим данную диаграмму, рисунок 1, для наших групп пользователей, а именно: врачи, пациенты и администраторы.

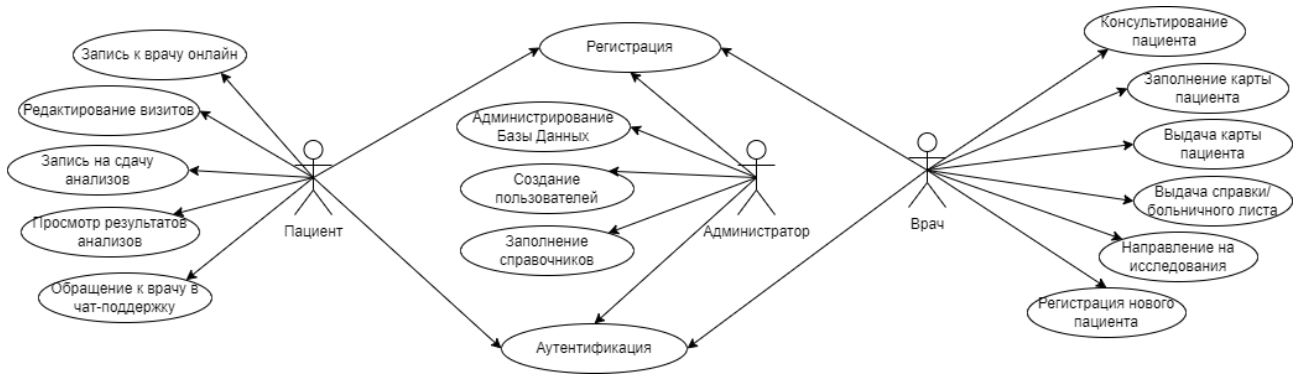


Рисунок 1 – Диаграмма вариантов использования

Среди основных возможностей программного продукта следует выделить:

- запись к врачу онлайн;
- просмотр своих записей;
- запись на анализы;
- просмотр результатов анализов онлайн;
- администрирование системы.

В связи с этим необходимо обозначить то, как должны работать будущие пользователи и как на их запросы должна реагировать система. Для отображения данной информации используем диаграмму последовательности, рисунок 2.

Диаграмма последовательности является одной из разновидности диаграмм взаимодействия и предназначена для моделирования взаимодействия объектов Системы во времени, а также обмена сообщениями между ними [3].

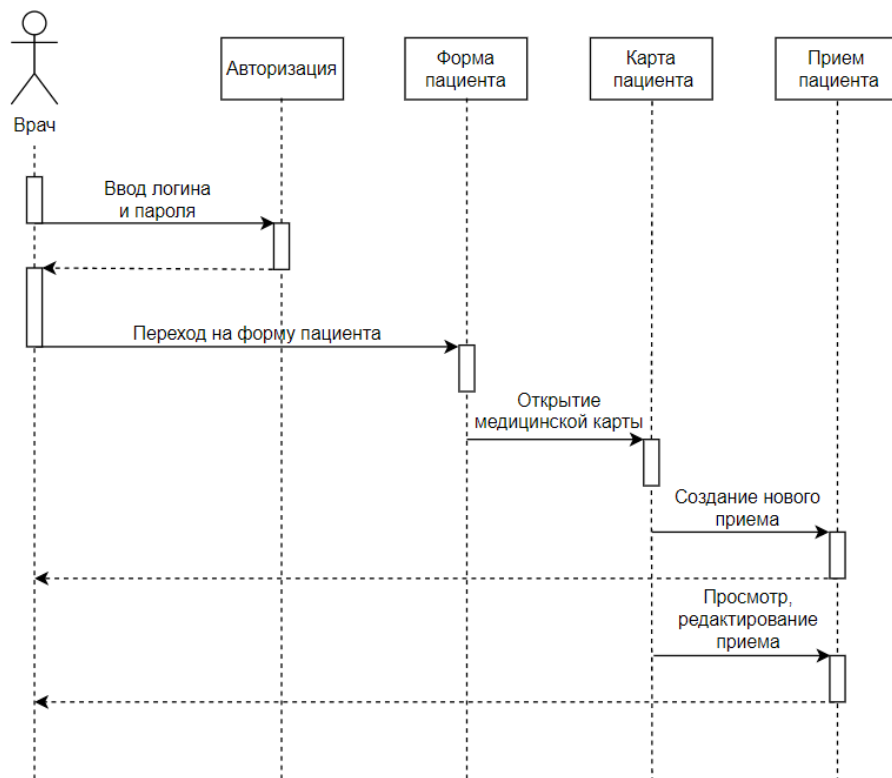


Рисунок 2 – Диаграмма последовательности

Следующим этапом следует выделить проектирование интерфейса новых модулей. Несомненно, стиль всего веб-приложения должен сохраняться во всех модулях. Результатом такого проектирования и разработки являются следующие доработки веб-приложения.

Доработана карта пациента, которую заполняет врач на приеме. Он указывает жалобы пациента, вносит рекомендации и назначения. Также на специальной схеме в виде человека имеется возможность отметить область, в котором у пациента возникают жалобы, рисунок 3.

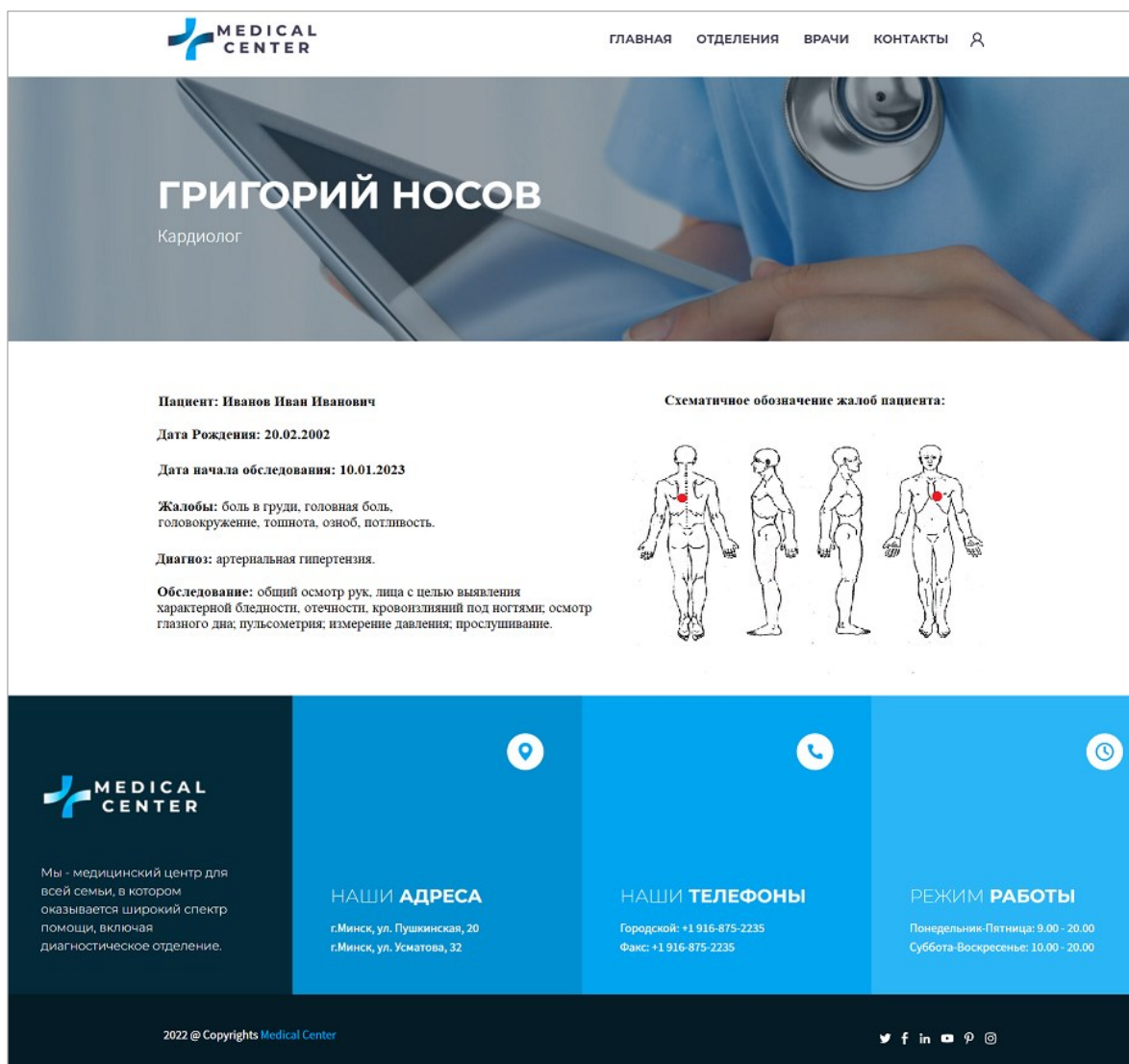


Рисунок 3 – Карта пациента

При указании предварительного диагноза врачу предлагаются рекомендованные анализы для его подтверждения, рисунок 4.

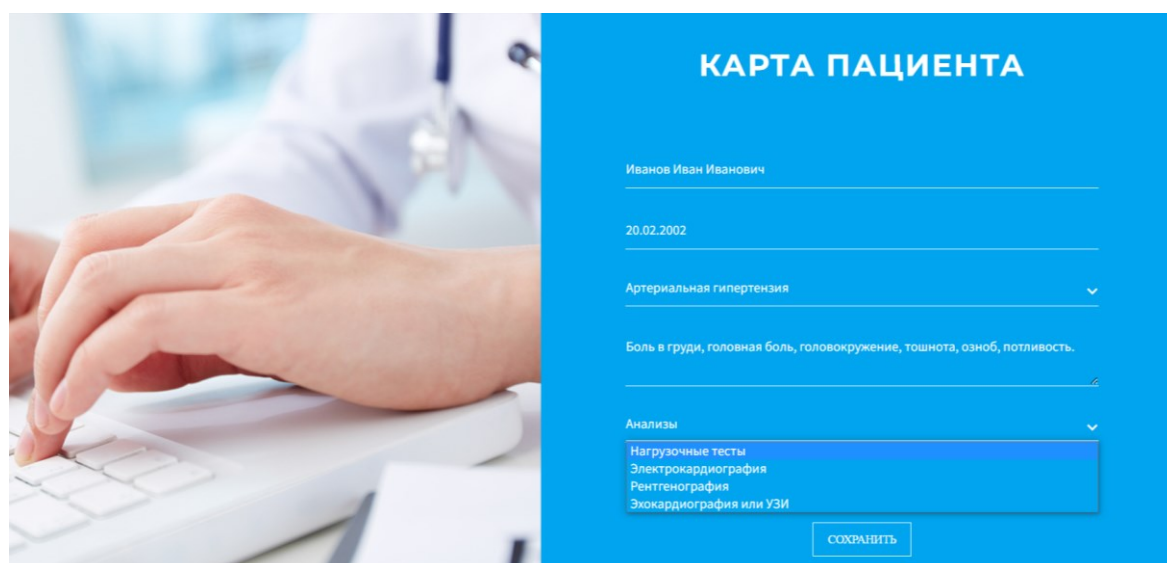


Рисунок 4 – Карта пациента

Пользователь-пациент может просмотреть свою медицинскую карточку онлайн. В данной карточке отображается информация, которую вносили врачи на приемах.

Также пациент может написать в чат-поддержку дежурному врачу и задать интересующий его вопрос, воспользовавшись формой, представленной на рисунке 5.

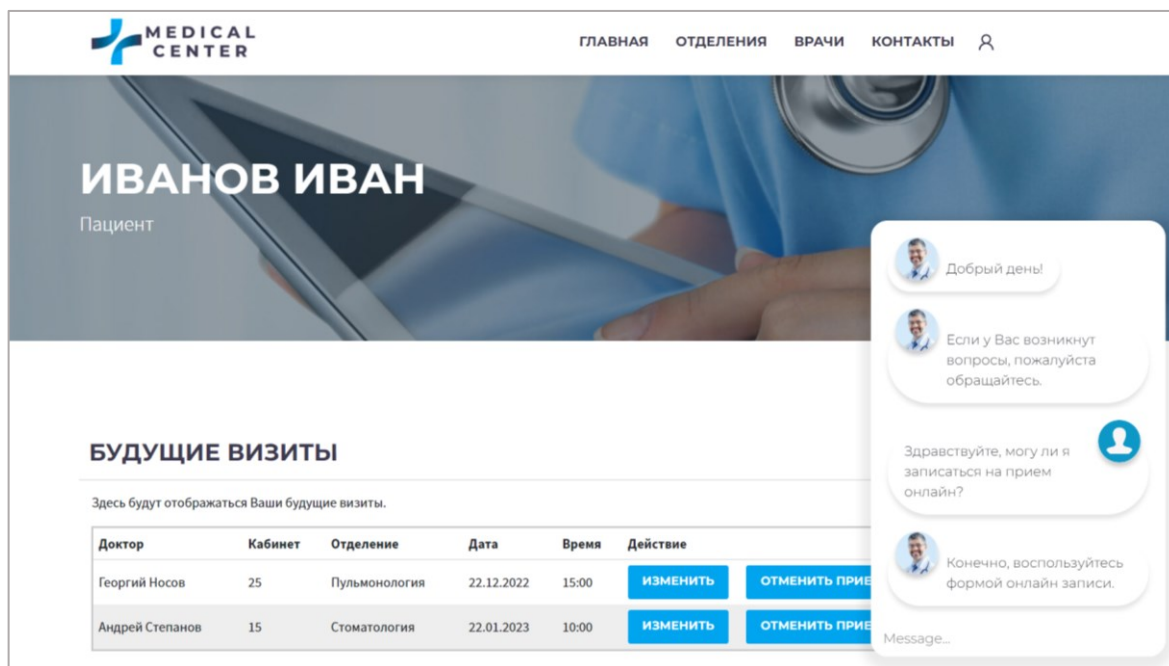


Рисунок 5 – Чат с доктором

Заключение. Разработанное веб-приложение для автоматизации работы медицинского центра предоставляет необходимую информацию, имеет такие возможности как формирование электронной медицинской карты пациента, онлайн запись к специалистам. Автоматизирует работу сотрудников, оформление медицинских документов, учет персонала, пациентов, обследований. Разработаны руководства пользователя и администратора. Перспективой развития является увеличение клиентской базы пользователей по географическому признаку.

Список использованных источников:

1. Лучшие научные исследования студентов и учащихся сборник статей Международной научно-практической конференции [Электронный ресурс] – Режим доступа : <https://elibrary.ru/item.asp?id=49875179>. – Дата доступа: 12.02.2023.
2. Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс] – Режим доступа : <https://habr.com/ru/post/566218/>. – Дата доступа : 01.03.2023.
3. Теория и практика UML. Диаграмма последовательности [Электронный ресурс] – Режим доступа : http://it-gost.ru/articles/view_articles/94. – Дата доступа : 05.03.2023

UDC 004.42

WEB APPLICATION WEB APPLICATION FOR AUTOMATION OF WORK MEDICAL CENTER

Koldunova A.A., Ladychuk I.S.

Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Bakunov A.M. – Master of Engineering Sciences

Annotation. The article deals with the issues of developing a software tool for automating the work of a medical center. A comparative analysis of existing solutions for various medical institutions is carried out, their advantages and disadvantages are revealed, examples of interface elements of the software being developed are given.

Keywords. Software, application development, medicine, medical service.

УДК 62-52

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ПОКРАСОЧНОЙ КАМЕРОЙ

Курилец А.С., студент

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

Шпак И.И. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В статье приведены результаты, полученные в процессе создания автоматизированной системы управления покрасочной камерой. Разработанная система осуществляет управление приточно-вытяжной вентиляцией окрасочно-сушильной камеры с поддержанием заданного подпора и требуемой температуры воздуха для выбранного режима работы. Осуществляется контроль уровня дозрывоопасных концентраций с применением специального сертифицированного датчика-газоанализатора. Предусмотрена возможность удаленного контроля через интерфейс RS-485. Автором статьи выполнено схемотехническое проектирование системы, разработаны алгоритм и программное обеспечение ее функционирования, а также конструкция печатного узла, с применением САПР.

Ключевые слова. покрасочная камера, автоматизированная система, микроконтроллер, контроль температуры, контроль уровня дозрывоопасных веществ, схемотехническое проектирование, разработка алгоритма и программного обеспечения, конструкторское проектирование.

Введение. Технологическим объектом автоматизации является покрасочная камера для покраски автомобильных кузовов. Для поддержания температурного режима и вентиляции камеры используется двухконтурная система, дополнительно оборудованная вентилятором, выполняющим роль вытяжки. Покрасочный комплекс обладает функцией рециркуляции, что позволяет экономить ресурсы для обогрева. Воздух в таком случае начинает циркулировать по замкнутому кругу. Использовать такой режим рационально при сушке нескольких свежеокрашенных деталей или всего авто.

Основная часть. Покрасочное помещение должно иметь развитую систему вентиляции. Засасываемый воздух проходит двойную степень фильтрации: перед вентилятором и перед попаданием в рабочее помещение. Попадающий воздух должен быть чистым, иначе теряется смысл использования покрасочной камеры – пыль и сор будут налипать на свежеокрашенные элементы. Фильтрующий элемент обязателен в местах отвода воздуха. В режиме рециркуляции в камеру должен подаваться чистый воздух. Очистка препятствует загрязнению окружающей среды, что также немаловажно. [1]

Автоматизированная система управления покрасочной камерой представляет комплекс аппаратных и программных средств для обеспечения полного контроля за температурным режимом, вентиляцией и концентрацией паров растворителей в пространстве камеры. Разработан микроконтроллерный блок управления сушильной камерой, который должен обеспечивать:

— автоматическую работу приточно-вытяжной вентиляции окрасочно-сушильной камеры с поддержанием заданного подпора и требуемой температуры воздуха для выбранного режима работы;

— контроль уровня дозрывоопасных концентраций;

— визуальную индикацию на дисплее текущего состояния системы, отображение сообщений, возможность удаленного контроля с ПК.

Технические характеристики разработанной автоматизированной системы приведены в таблице 1.

Таблица 1 – Технические характеристики автоматизированной системы управления покрасочной камерой

Параметр	Значение
Контролируемый диапазон температуры в камере	20...100 °C±1°C
Диапазон контролируемого перепада давления воздуха в системе вытяжной вентиляции	0...500 Па ± 1 Па
Контроль уровня дозрывоопасных концентраций	0...50 % НКПР
Аналоговые входы	4...20 мА, (2x0-10 В)
Дискретные выходы	2 А (~220 В/50 Гц)
Индикация	LED дисплей
Интерфейс	RS-485
Питание – сеть 220 В ±15%, 50Гц.	сеть 220 В ±15%, 50Гц.

Одним из важнейших параметров для газоанализаторов является НКПР – это нижний концентрированный предел распространения пламени. Данная аббревиатура обозначает минимальную концентрацию горючего вещества в смеси с окислителем, при которой пламя может распространиться на любое расстояние от очага возгорания. В качестве горючего вещества выступают пары растворителей лакокрасочных материалов. Для контроля НКПР применяются специальные датчики газоанализаторы концентрации паров. Уровень НКПР регламентирован в ТКП 474-2013 [2].

Микроконтроллерный блок сохраняет свои параметры в пределах норм, установленных техническим заданием, стандартами или техническими условиями, после и в процессе воздействия климатических факторов, значения которых установлены ГОСТ 15150-69. Устройство устойчиво к воздействию относительной влажности воздуха до 80 % при температуре 25 °С, до 95 % при температуре 35 °С.

Разработанная система рассчитана для эксплуатации в помещениях с искусственно регулируемые климатическими условиями, например, в закрытых отапливаемых или охлаждаемых и вентилируемых производственных и других помещениях (с отсутствием воздействия атмосферных осадков, прямого солнечного излучения, ветра, песка, пыли, наружного воздуха, отсутствие или существенное уменьшение воздействия рассеянного солнечного излучения и конденсации влаги), а конкретнее - в лабораториях, капитальных жилых и других подобного типа помещениях. Следовательно, проектируемое устройство относится к категории исполнения 4.2. Микроконтроллерный блок управления должен соответствовать климатическому исполнению УХЛ 4.2.

Требования к конструкции микроконтроллера блока управления автоматизированной системы покрасочной камеры вытекают из его функционального назначения и условий его эксплуатации. Конструкция блока должна обеспечивать ремонтпригодность, удобство в эксплуатации, иметь, по возможности малые габариты и вес, и высокую надежность в работе. Эстетические требования должны соответствовать ГОСТ 23852 – 79. Конструкция прибора должна отвечать требованиям к технологичности по ГОСТ18831 – 73 и ГОСТ 14205 – 83.

Управление осуществляется согласно управляющей программы микроконтроллера. В программу заложен алгоритм ПИ-регулятора температуры с ШИМ выходом. Один дискретный релейный выход формирует закрывающие импульсы на сервопривод горелки калорифера, а второй – открывающие. В зависимости от вида нагревателя калорифера может использоваться и один управляющий дискретный выход, если нагрев осуществляется электрическими ТЭНами. Предусматриваются любые варианты исполнения системы нагрева воздуха в камере. Дискретные выходы для управления исполнительного механизма заслонки формируют сигналы по сигналу датчика перепада давления на вытяжном вентиляторе. При превышении концентрации паров к предельному значению 50 % НКПР происходит полное открытие воздушной заслонки. При нулевом перепаде давления формируется сигнал аварийного отключения горелки и передается соответствующее тревожное сообщение на диспетчерский ПК.

Для реализации всех перечисленных функций схема электрическая структурная системы будет иметь вид, приведённый на рисунке 1.

Система управления автоматизированной покрасочной камеры состоит из 12 основных узлов. Все узлы непосредственно связаны лишь с блоком управления.

Блок управления осуществляет сохранение заданной температуры покрасочной камеры, измеряет и обрабатывает сигналы от датчиков и клавиатуры, формирует команды остальным узлам системы управления. В качестве основы блока управления используется 8-битный микроконтроллер.

Дисплей отображает графическую информацию для пользователя. Данный дисплей выводит необходимую информацию о режиме работы покрасочной камеры, значения текущей и заданной температуры, давления, концентрации.

С помощью клавиатуры пользователь задает температурную уставку и производит запуск и остановку оборудования камеры: приточно-вытяжных вентиляторов и горелки. Возможно их отдельное включение или отключение по выбору пользователя. Данная клавиатура состоит из трех кнопок. Этого достаточно для использования кнопок в качестве навигации по меню и ввода значений путем многократного нажатия.

Датчик температуры служит в качестве элемента обратной связи для регулирования температуры внутри покрасочной камеры. Регулировка осуществляется горелкой, водяным или электрическим теплообменником, в зависимости от желания и потребностей заказчика. Будет использоваться датчик температуры с встроенным преобразователем с унифицированным выходным сигналом 0...10 В и точностью не ниже ± 1 °С.

Датчик давления воздуха формирует аналоговый сигнал 0...10 В, пропорциональный перепаду давления 0...500 Па в вентиляционном канале.

Датчик газоанализатора – необходим для выявления токсичных веществ в атмосфере и их концентрацию по сравнению с данными предельно допустимых значений этих веществ. В качестве этого датчика выбирается специальный сертифицированный датчик концентрации горючих паров растворителей с унифицированным аналоговым выходом 4...20 мА.

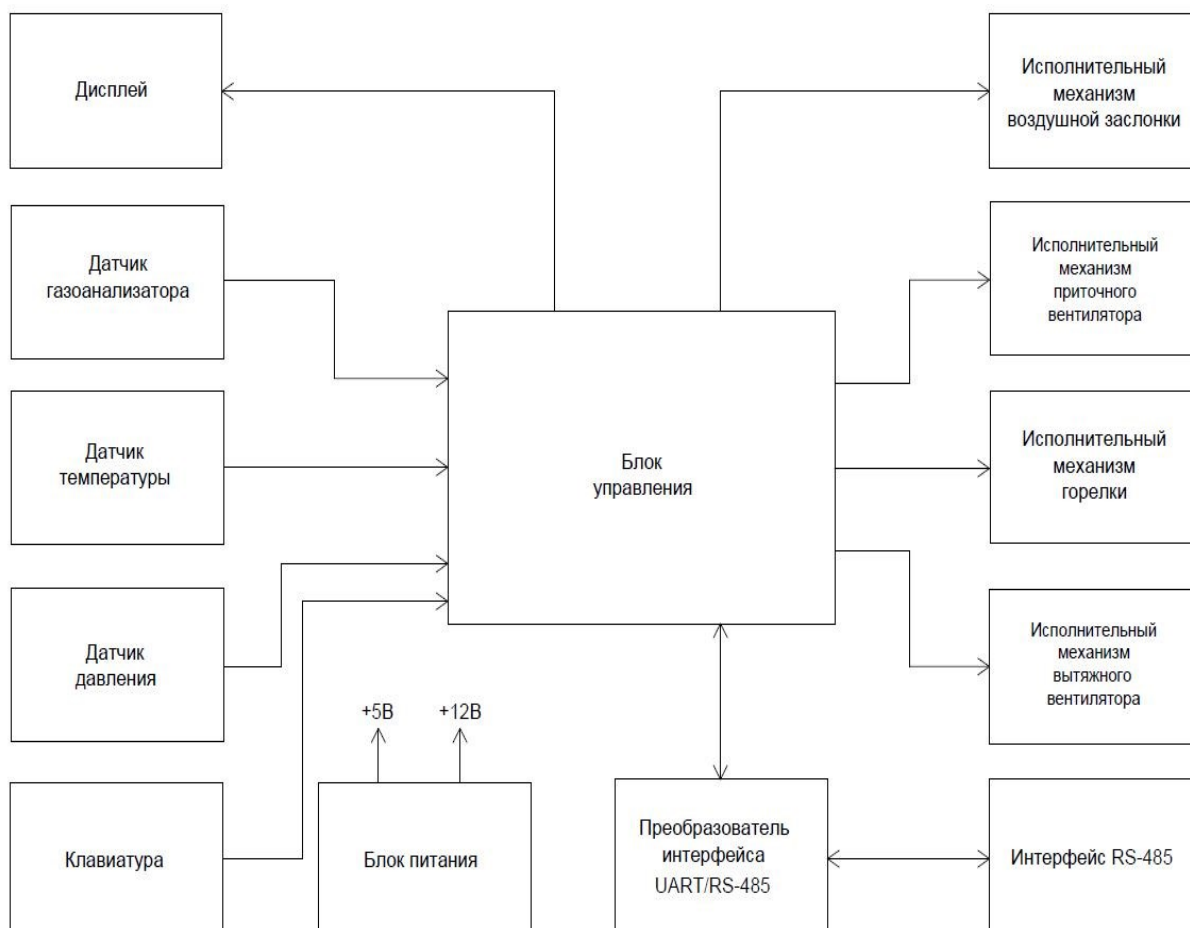


Рисунок 1 – Схема электрическая структурная автоматизированной системы управления покрасочной камерой

Преобразователь интерфейса UART/RS-485 служит для согласования последовательного асинхронного приемопередатчика микроконтроллера с интерфейсом RS-485. RS-485 является наиболее распространенным для связи вторичных приборов систем автоматизации с ПЛК или промышленными компьютерами. Для подключения к USB ПК или Ethernet могут применяться любые преобразователи интерфейсов различных фирм-производителей.

Интерфейс RS-485 реализуется на основе микросхемы помехозащищенного трансивера UART/RS-485 рассчитанного для работы в неблагоприятных промышленных условиях. Шинные выводы этих устройств устойчивы к высоким уровням кратковременных выбросов напряжения и электростатическому разряду (ESD) по IEC, что исключает необходимость в дополнительных защитных компонентах на системном уровне.

Для проекта выбран микроконтроллер с встроенным многоканальным АЦП. Многоканальность означает, что на входе единственного модуля АЦП установлен аналоговый мультиплексор, который может подключать этот вход к различным выводам микроконтроллера для осуществления измерений нескольких независимых аналоговых величин с разнесением по времени. Входы мультиплексора могут работать по отдельности (в несимметричном режиме для измерения напряжения относительно "земли") или (в некоторых моделях) объединяться в пары для измерения дифференциальных сигналов. Иногда АЦП дополнительно снабжается усилителем напряжения с фиксированными значениями коэффициента усиления 10 и 200. Разрешающая способность АЦП в микроконтроллерах AVR \square 10 двоичных разрядов, чего для большинства типовых применений достаточно.

ATmega324PA используется для встраиваемых, недорогих приложений реального времени. Микроконтроллер имеет высокое соотношение производительность/потребление, максимальную тактовую частоту 20 МГц весь необходимый набор аппаратных модулей. Имеется выбор программных средств разработки для данных микроконтроллеров и примеров применения. Поэтому выбор остановлен конкретно на этом микроконтроллере. [3]

ATmega324PA это 8-разрядный AVR-микроконтроллер с внутрисистемно-программируемой Flash памятью размером 32 Кбайт.

Основные характеристики ATmega324PA:

- 32Кб Flash память программ;
- 2 Кб SRAM памяти;
- 1 Кб EEPROM;
- питание 1,8...5,5 В;

- 8 x 10-бит АЦП;
- 32 входов/выходов совместимых с 5 В;
- часы реального времени;
- 1 watchdog таймер;
- I2C – 1 шина;
- UART – 2 шины;
- SPI – 1 шина;
- таймер 8 бит – 2;
- таймер 16 бит – 1;
- таймер ШИМ – 6 каналов.

Заключение. Отличительная особенность и преимущество предлагаемой разработки заключается в том, что она представляет собой специализированное компактное программируемое устройство, которое выполняет полный контроль и управление покрасочной камерой в автоматическом режиме. Для решения подобных задач в настоящее время для управления данным процессом необходимо применять несколько различных приборов или более дорогостоящие решения на базе программируемых логических контроллеров.

Новая автоматизированная система осуществляет управление приточно-вытяжной вентиляцией окрасочно-сушильной камеры с поддержанием заданного подпора и требуемой температуры воздуха для выбранного режима работы. Осуществляется контроль уровня взрывоопасных концентраций с применением специального сертифицированного датчика-газоанализатора. Дисплей блока управления обеспечивает наглядную индикацию текущего состояния системы, отображение сообщений. Предусмотрена возможность удаленного контроля через интерфейс RS-485.

Конструкция блока управления была реализована в виде небольшого блока на основе выбранной современной элементной базы, и проведённого электрического расчёта всех функциональных узлов. Поэтапно были разработаны алгоритм работы блока управления и программное обеспечение [4] для микроконтроллеров.

Топология печатной платы и печатного узла на её основе были спроектированы с использованием системы автоматизированного проектирования Altium Designer.

Применение результатов, полученных в процессе разработки системы управления, при изготовлении, монтаже и наладке покрасочных камер на территории заказчика ООО «Профикарлайн» позволило сократить затраты на производство систем управления покрасочных камер [5].

Список использованных источников:

1. Комбинированные окрасочно-сушильные камеры [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.matrixplus.ru/index1-014.htm> (дата обращения: 02.11.2022).
2. ТКП 474-2013 Категорирование помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.
3. ATmega 324A/324PA [Электронный ресурс]. – Режим доступа: <Datasheet/Atmel/2048001.pdf> Дата доступа: 02.04.2023.
4. Языки программирования микроконтроллеров [Электронный ресурс]. – Режим доступа: <https://mcscru.ru/index.php/soft/42-lmcsu/67-programmlang>. Дата доступа: 02.04.2023.
5. Справка о внедрении результатов дипломного проекта на тему «Автоматизированная система управления покрасочной камерой» студента факультета компьютерных технологий ИИТ БГУИР Курильца Артема Сергеевича в производство на ООО «Профикарлайн».

UDC 62-52

AUTOMATED CONTROL SYSTEM OF THE PAINTING CHAMBER

Kurilec A.S.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Shpak I.I. – Candidate of Engineering Sciences, Associate Professor

Annotation. The article presents the results obtained in the process of creating an automated control system for the paint chamber. The developed system controls the supply and exhaust ventilation of the paint-drying chamber with the maintenance of a given backstop and the required air temperature for the selected operating mode. The level of pre-explosive concentrations is monitored using a special certified gas analyzer sensor. The possibility of remote control via the RS-485 interface is provided. The author of the article performed circuit design of the automated system, developed an algorithm and software for its operation, as well as the design of the printed unit of the control unit, using modern CAD.

Keywords. Painting chamber, automated system, microcontroller, temperature control, explosive level control, circuit design, algorithm and software development, design engineering.

УДК 004.42+ 331.108.2

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА РАБОЧИХ КАДРОВ ПРЕДПРИЯТИЯ

Самсонов Н.Н., студент

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

Кунцевич О.Ю. – канд. пед. наук, доцент каф. ИСиТ

Аннотация. Статья посвящена разработке программного средства для автоматизации учета рабочих кадров предприятия на платформе .NET, представлен пример элемента пользовательского интерфейса приложения, обоснован выбор средств разработки, представлены некоторые особенности этапов разработки приложения.

Ключевые слова. Разработка программного обеспечения, автоматизация, учет сотрудников.

Введение. Проблема автоматизации производственных процессов и процессов управления как средства повышения производительности труда всегда являлась и остается актуальной.

Основная работа специалиста по кадрам заключается в подборе персонала. Специалист должен правильно оценить возможности человека и предложить ему подходящую должность. Основной задачей при реализации разработанного программного средства является информационная поддержка сотрудников отдела кадров.

Проект ориентирован на реализацию, в частности, таких функциональных особенностей разрабатываемого приложения как ведение базы данных (содержащей информацию о сотрудниках), выполнение статистики кадровых операций, формирование документов и отчетов, ведение журнала рабочего времени. Для публикации отчетов реализованы процессы импортирования данных в стороннее программное обеспечение – Microsoft Excel и Microsoft Word.

Для реализации проекта необходимо решить ряд задач, в частности:

- проанализировать существующие аналогичные программные продукты для автоматизации кадрового процесса;
- разработать проектную документацию для программного средства (далее – ПС);
- спроектировать интерфейс приложения средствами создания меню, кнопочных форм, панелей инструментов;
- осуществить ведение базы данных, содержащей информацию о сотрудниках, предоставляемую при приеме на работу;
- разработать алгоритмы работы ПС и выполнить их программную реализацию;
- протестировать разработанное ПС;
- разработать руководство по использованию ПС.

Определим назначение и возможности разрабатываемого ПС:

- оптимизация работы с данными о сотрудниках предприятия;
- осуществление автоматизированной обработки данных по приёму кадров, формированию кадровых документов, табеля учёта рабочего времени;
- обеспечение возможности получения статистической отчётности о работающих сотрудниках;
- осуществление возможности просмотра и поиска необходимой пользователю информации.

В качестве среды разработки программного средства выбрана среда разработки Microsoft Visual Studio, языка разработки – язык C#, системы управления базами данных – SQL Server и среда Management Studio [1-4].

Основная часть. Перейдем к описанию некоторых этапов разработки.

Важнейшим компонентом процесса создания сложных систем и программного обеспечения, в частности, является этап системного анализа и моделирования соответствующей предметной области. Данный этап является предпроектным. Его цель заключается в разработке спецификации проекта.

Одним из инструментов системного анализа является CASE-средство ERwin Data Modeler (ERwin) [5].

В данной программе нами построена контекстная диаграмма IDEF0 для проекта «Программное средство для автоматизации учета рабочих кадров предприятия» (рисунок 1), а также диаграммы декомпозиции первого и второго уровней.

Концептуальный прототип приложения представляет собой описание внешнего пользовательского интерфейса – систему меню, диалоговых окон и элементов управления.

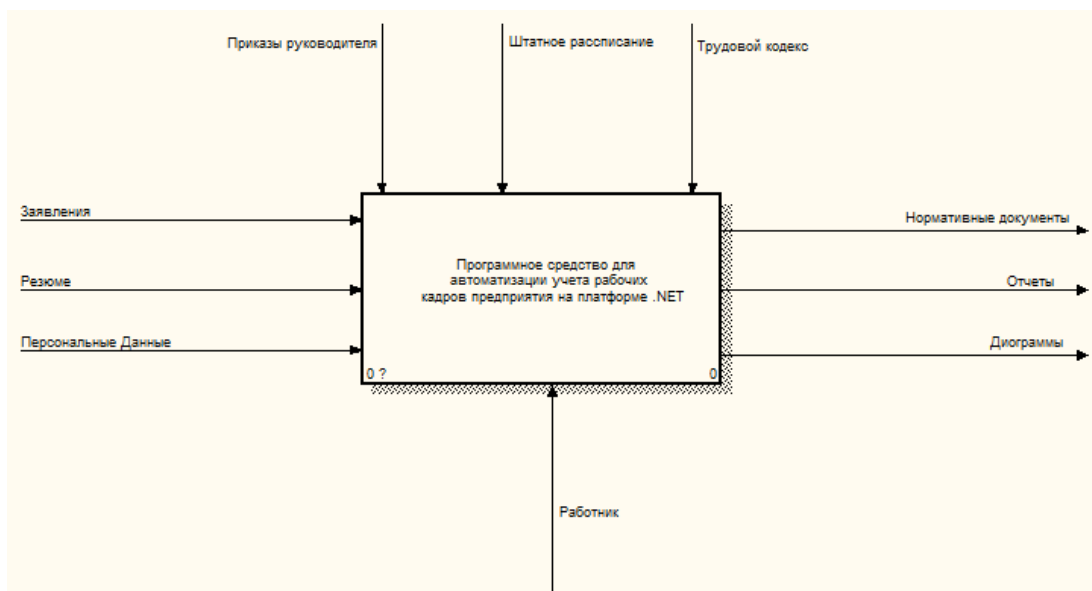


Рисунок 1 – Контекстная диаграмма IDEF0 для «Программное средство для автоматизации учета рабочих кадров предприятия»

Все действия можно будет выполнить с помощью главной формы, которая появляется после запуска ПС. Главная форма будет иметь кнопочное меню программы, позволяющее вызывать остальные модули программы, осуществлять различные манипуляции с данными, а также получать справочную информацию о работе с приложением. Используя меню, пользователь имеет возможность завершить работу всего приложения с сохранением последних изменений.

Рассмотрим предложенный концепт интерфейса более подробно:

- нажав на иконку в правой верхней части экрана приложение можно свернуть или закрыть;
- в левой части экрана отображаются кнопки навигации по различным компонентам программы;
- при нажатии на кнопку «Сотрудники» будет отображена форма, которая хранит всю информацию о сотрудниках; также возможно открыть форму для заполнения данных о сотруднике;
- при нажатии на кнопку «Формирование документов» будет отображена форма, в которой находится выбор документов для экспорта;
- при нажатии на кнопку «Отчёты» будет отображена форма, в которой находится отчётность о работающих сотрудниках.

Для формирования отчета необходимо сгенерировать нужные данные, выбрав соответствующий пункт в меню (рис. 2).

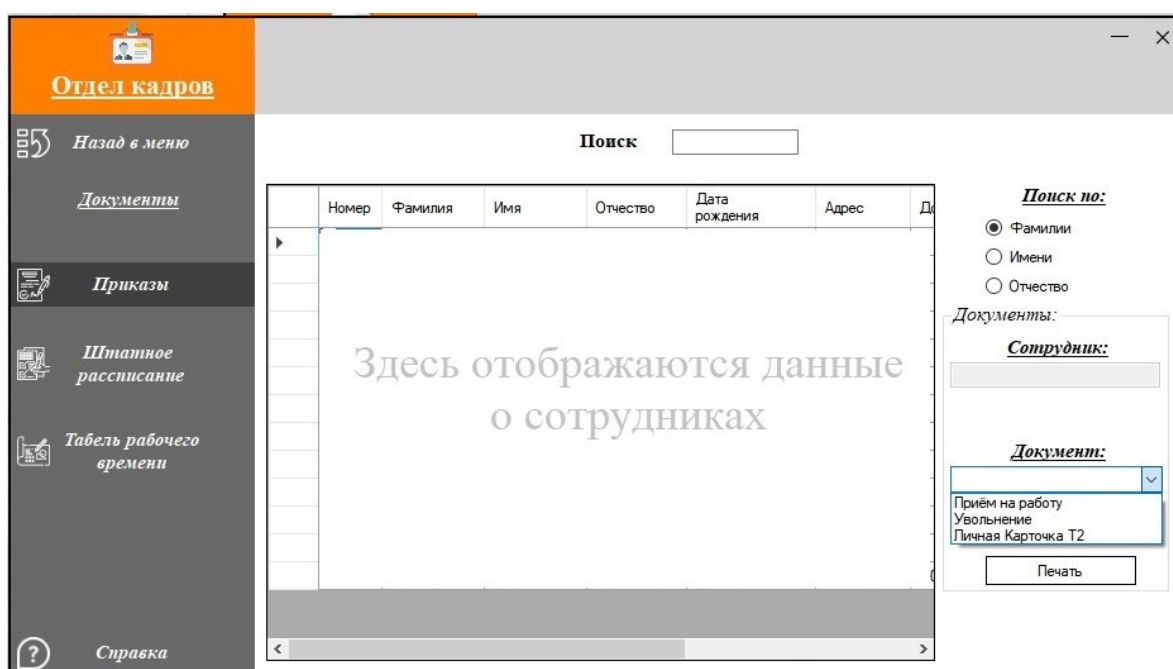


Рисунок 2 – Страница выбора документов для экспорта данных и формирования отчета

Основное меню программы будет представлено на главной форме, на которой находятся основные элементы управления программой. С помощью кнопок на главной форме можно получить доступ к информации о добавленных ранее сотрудниках и добавить информацию о новых сотрудниках.

Для описания механизмов работы модулей программы были разработаны, в частности, алгоритм работы программного средства, алгоритм создания данных о сотруднике, диаграмма вариантов использования др.

В результате тестирования программы были проверены все возможные действия, навигация по пунктам меню, которые может произвести пользователь. В частности, были протестированы следующие функции:

- добавление сотрудника;
- удаление сотрудника;
- запуск справочной системы;
- создание отчета «О принятых сотрудниках»;
- создание документа «Штатное расписание»;
- создание документа «Табель рабочего времени»;
- получение информации о сотрудниках, у которых истекает трудовой договор и др.

После все собранные и найденные ошибки в работе программного средства были устранены.

Заключение. Таким образом, разработанное приложение будет способствовать автоматизации формирования кадровых документов, оптимальной обработке информации о сотрудниках предприятия, проведению учёта движения кадров, включать в себя возможность экспорта данных в иные приложения. В приложении реализованы следующие функции: ведение штатного расписания предприятия, создание и ведение базы данных сотрудников, оформление и ведение личной карточки сотрудника (форма Т-2), оформление приказов о принятии и увольнении работника, ведение табеля рабочего времени, формирование списка сотрудников, у которых заканчивается контракт.

Проект находится в рабочем состоянии и готов к использованию. Также имеется возможность дальнейшего усовершенствования данного программного средства. Приложение может применяться в любой организации независимо от количества сотрудников, может модернизироваться и дополняться в зависимости от изменений в структуре организации, реальных задач, не требует больших технических ресурсов компьютерной техники.

Список использованных источников:

1. Шумаков, П. В. Программирование на языке Transact SQL при работе с СУБД Microsoft SQL Server / П. В. Шумаков. – М. : АСУ-61, 2019. – 772 с.
2. Документация по Visual Studio [Электронный ресурс]. – Режим доступа: <https://visualstudio.microsoft.com/ru/>.
3. Документация по C# [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/>.
4. Документация по SQL Server Management Studio [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16>.
5. Маклаков, С.В. ВРwin и Erwin. Case-средства разработки информационных систем / С.В. Маклаков. – М.: Диалог-МИФИ, 2001. – 306 с.

UDC 004.42+331.108.2

SOFTWARE TOOL FOR AUTOMATION OF STAFF ACCOUNTING OF AN ENTERPRISE

Samsonov N.N.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus*

Kuntsevich O.Yu. – Candidate of Pedagogic Sciences, Associate Professor

Annotation. The work is devoted to the development of a software tool for automating the accounting of enterprise workforce on the .NET platform, the concept of the user interface of the application is presented, the choice of development tools is justified, the advantages and disadvantages are listed. C# was chosen as the programming language.

Keywords. Software development, automation, employee accounting..

УДК 681.58'8

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПЕРВИЧНОЙ ОБРАБОТКИ РАДИОЛОКАЦИОННОЙ ИНФОРМАЦИИ

Сапега Я.Е., студент

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

Капанов Н. А. – маг. техн. наук, ст. препод. каф. ИСиТ

Аннотация. Работа содержит описание модуля первичной обработки информации в радиолокационных системах обнаружения объектов по азимуту и углу места с передачей информации на видеотерминалы автоматизированных рабочих мест.

Ключевые слова. Автоматизированная система, радиолокационные цели, азимут, угол места, обработка информации, вращающийся трансформатор, сельсин датчик, положение антенны, радиолокационная информация, потенциометрический датчик, высотомер.

Введение. Главной задачей модуля первичной обработки информации является возможность сопряжения с радиолокационной станцией (РЛС) и подвижным радиовысотометром (ПРВ), дальнейшая обработка и передача информации. Для реализации задач сопряжения, обработки и синхронизации, модуль первичной обработки информации оснащен модулем первичной обработки радиолокационной информации (РЛИ) и модулем ПРВ по азимуту и углу места [1]. Эти модули предназначены для обработки РЛИ от РЛС и ПРВ, и для исполнения команд управления наземными радиолокационными запросчиками (НРЗ) и ПРВ.

Основная часть. Структурная схема модуля представлена на рисунке 1.

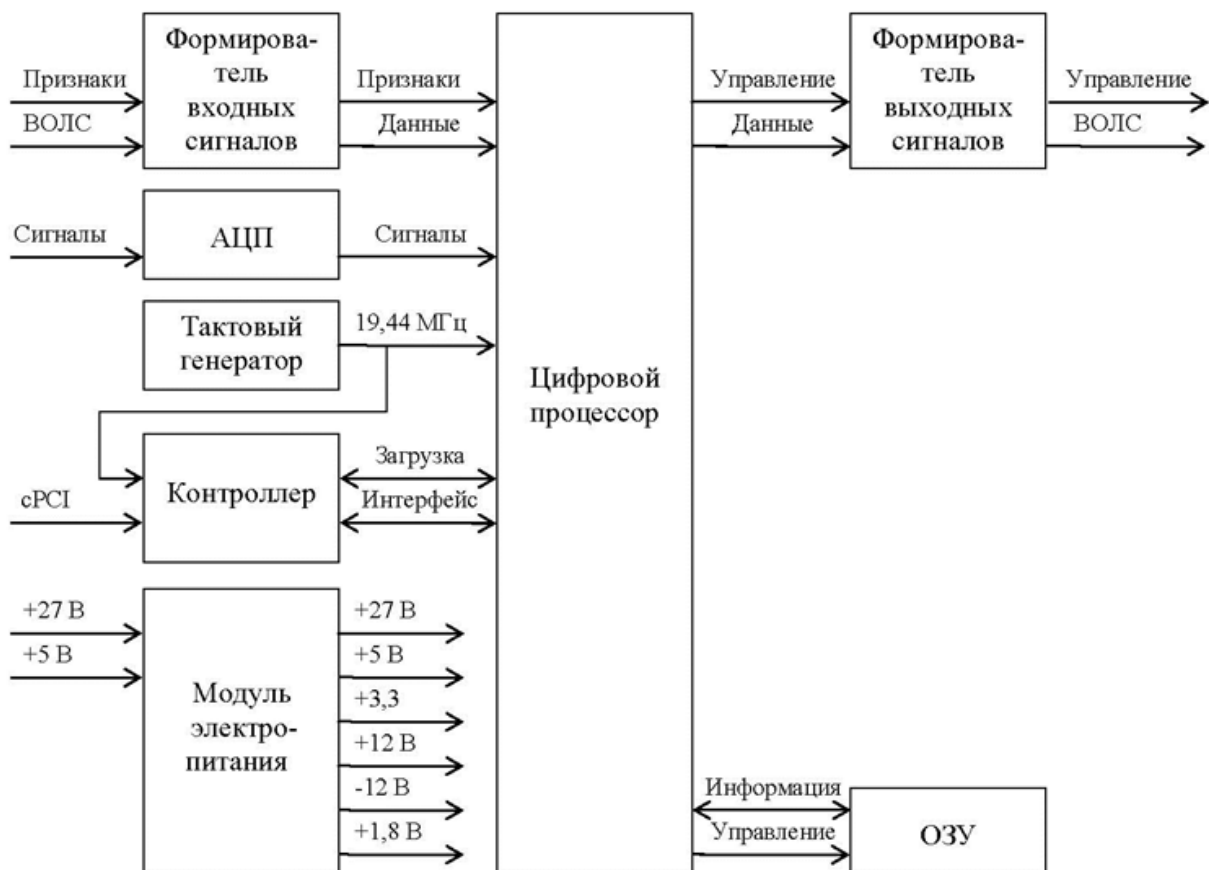


Рисунок 1 – Схема модуля первичной обработки РЛИ

Модуль выполнен на основе цифрового процессора с дополнительными функциональными частями, обеспечивающими его электропитание, конфигурирование, хранение оперативной информации и интерфейсы ввода/вывода.

Цифровой процессор выполнен на базе программируемой логической интегральной схемы и обеспечивает прием, обработку, преобразование и вывод информации в центральный процессорный блок модуля первичной обработки информации и дальнейшей коммутации с передающим модулем.

В формирователе входных сигналов обеспечивается прием и нормирование по амплитуде входных сигналов, несущих информацию о признаках режимов работы и синхронизации СРЛ, и сигналов, поступающих по волоконно-оптической линии связи (ВОЛС).

В шестиканальном аналого-цифровом преобразователе (АЦП) РЛИ преобразуется в 12-разрядные цифровые коды для дальнейшей обработки в цифровом процессоре.

Оперативное запоминающее устройство (ОЗУ) предназначено для функций записи, хранения и чтения обрабатываемой РЛИ. За счет работы контроллера обеспечивается следующий функционал:

- сопряжение с процессорной платой центрального процессорного блока 32-разрядными сообщениями с использованием интерфейса Compact PCI;
- программирование и реконfigurирование цифрового процессора.

Интерфейс информационного взаимодействия контроллера с цифровым процессором обеспечивает ввод/вывод данных по 32-разрядной двунаправленной шине данных.

Формирователь выходных сигналов предназначен для формирования сигналов управления, а также для трансляции РЛИ по ВОЛС далее в передающие модули, и обеспечивает:

- формирование выходных импульсных сигналов;
- защиту выходных цепей от перегрузок;
- преобразование электрических сигналов в оптические.

Для обеспечения взаимодействия всех функциональных частей модуля используется высокостабильный тактовый генератор, работающий на частоте 19,44 МГц.

Рассмотрим работу модулей синхронизации с РЛС и ПРВ по азимуту и углу места. Структурная схема данных модулей представлена на рисунке 2.



Рисунок 2 – Схема модуля синхронизации с РЛС и ПРВ по азимуту и углу места

За счет работы данных модулей обеспечивается:

- сопряжение изделия с датчиками положения антенны РЛС и ПРВ и формирование команд управления угловым положением антенны ПРВ [1];
- передачу на модули первичной обработки информации угловой информации о положении антенн РЛС и ПРВ;
- сопряжение с центральным процессорным устройством модуля первичной обработки информации.

Контроллер датчиков положения антенны обеспечивает:

- прием сигналов от четырех сельсин-датчиков или датчиков синусно-косинусных вращающихся трансформаторов [2] и формирование по каждому каналу 14-разрядного цифрового кода угла поворота ротора датчика. Для двух каналов предусмотрена возможность задания режима 10-разрядного преобразования;

— формирование 10-разрядных цифровых кодов мгновенных значений амплитуд синусной и косинусной составляющих напряжения и опорного напряжения, поступающих на входы преобразователей угол-код;

цифровое управление коэффициентами передачи амплитуд сигналов, поступающих на входы преобразователей угол-код, а также управление сдвигом фазы сигнала опорного напряжения по каждому из четырех каналов.

Передача в цифровой процессор данных, содержащие коды амплитуд сигналов, и управление коэффициентами передачи со стороны цифрового процессора выполняются по последовательному интерфейсу.

Контроллер интерфейса оператора высотомера обеспечивает:

прием сигналов от потенциометрического датчика высоты и формирование 14-разрядного цифрового кода постоянного напряжения, поступающего от потенциометрического датчика высоты;

прием сигналов квитирования и выдачу команд управления оператором высотомера.

Контроллер управления положением антенны обеспечивает формирование двух напряжений рассогласования по цифровым 12-разрядным кодам напряжения рассогласования, поступающим от цифрового процессора.

Последним обязательным технологическим устройством модулей первичной обработки информации будут являться модули отображения аналоговой РЛИ на АРМ. За счет данных модулей обеспечивается следующий функционал:

формирование информации для отображения совмещенной аналоговой и цифровой РЛИ от РЛС, НРЗ и ПРВ на мониторах АРМ;

сопряжение с центральным процессорным устройством и видеоконтроллером центрального процессорного блока;

прием РЛИ по оптическому кабелю от модулей первичной обработки РЛИ.

Помимо модулей отображения аналоговой РЛИ на АРМ необходима коммутации с терминалами навигационных измерений (ТНИ). Данные терминалы предназначены для приема и обработки радиосигналов глобальных навигационных спутниковых систем (ГНСС) NAVSTAR (США) и ГЛОНАСС (Россия), определения координат местоположения изделия и синхронизации текущего времени в центральном процессорном блоке с системой единого времени. Данный терминал состоит

из антенного блока, конструктивно объединяющего антенну и предварительный радиочастотный усилитель, и приемного блока, который предназначен для обработки информации принятой антенным блоком и выдачи результатов обработки по последовательному интерфейсу внешним потребителям. Антенный блок представляет собой микрополосковую антенну круговой поляризации с встроенным маломощным усилителем и предназначенную для приема радиосигналов ГНСС.

Заключение. Таким образом, при таком мультимодульном иерархическом построении мобильной автоматизированной системы выдачи радиолокационных целей в воздушном пространстве, будет обеспечен прием всей необходимой цифровой и аналоговой информации от всех объектов, с последующей обработкой и оперативной связью с вышестоящими органами за счет проводных или беспроводных каналов связи.

Список использованных источников:

1. Гринкевич А. В., *Радиолокация: учеб. пособие* / А. В. Гринкевич — Минск: БГУИР, 2015. — 190 с.
2. Волков Н.И., Миловзоров В.П. *Электромашинные устройства автоматики: учеб. пособие* / Н.И. Волков — Москва: Высшая школа, 1986 — 335 с.

UDC 681.58'8

AUTOMATED SYSTEM FOR PRIMARY PROCESSING OF RADAR INFORMATION

Sapega Y. E.

*Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Kapanov N. A. — Master of Engineering Sciences

Annotation. The work contains a description of the module for the primary processing of information in radar systems for detecting objects in azimuth and elevation with the transfer of information to video terminals of automated workstations.

Keywords. Automated system, radar targets, azimuth, elevation angle, data processing, rotary transformer, synchro sensor, antenna position, radar information, potentiometric sensor, altimeter..

УДК 004.75+004.91

РАСПРЕДЕЛЕННАЯ СИСТЕМА СТАТИСТИЧЕСКОГО АНАЛИЗА ДОКУМЕНТОВ

Северин К. М., студент

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

Парамонов А.И. – канд. техн. наук, доцент, зав. каф. ИСиТ

Аннотация. Рассмотрены методы анализа текста. Выполнен анализ задач, решаемых статистическим методом. Изучена проблема автоматической классификации текстовых документов. Рассмотрен алгоритм TextRank для классификации документа. Предложено проектное решение информационно-аналитической системы классификации документов.

Ключевые слова. Анализ документов, статистический анализ, TextRank, алгоритм, классификация текста, информационно-аналитическая система, граф, распределенная система, сервер, клиент.

Введение. Значительный рост объемов цифровой информации, в том числе цифровых документов, определяет потребность в новых инструментах для их обработки. Зачастую это массивы неструктурированных текстовых документов, по которым необходимо выполнять поиск, осуществлять их систематизацию, проводить оценку и многие другие операции. Очевидно, что для всех этих задач необходимо выполнить анализ цифрового текста.

В настоящее время разработано и используется множество разных подходов и методов анализа текстовых документов, которые направлены на решение различных прикладных задач: интент-анализ, контент-анализ, фоносемантический, графематический, морфологический, синтаксический, семантический и другие виды анализа [1].

Основная часть. Зачастую для достижения лучших показателей при обработке текста любому качественному анализу должен предшествовать количественный. Стоит отметить, что применение статистики для анализа текстов – традиционная задача. На сегодняшний день существует несколько методов статистического анализа текста. Среди них выделяют процедуры количественных исследований, частотный анализ, контент-анализ, ранжирование данных и другие. С помощью статистического анализа можно решить множество задач:

- классификация текста (статистическая стилистика);
- установление авторства (проведения атрибуции текстов на основании неповторимого сочетания статистических параметров авторского текста);
- описание поведения языковых единиц (букв, морфем, слов) в тексте, их распределение, сочетаемость, частота употребления;
- измерение информативности (расчет количества информации, содержащейся в тексте и его составных частях);
- восстановление текстов (описание структуры на основании очень ограниченной исходной информации) и т.д.

Наибольший интерес вызывает применение статистического анализа для решения задачи ранжирования документов по темам, отраслям и т.п. Ранжирование (или категоризация) документов – это частный случай классификации. Ее задача состоит в том, чтобы отнести объект к некоторой категории. В свою очередь, классификация является одной из основных задач компьютерной лингвистики, поскольку к ней сводится ряд других.

Задачу автоматической классификации текстовых документов можно решить с помощью разных алгоритмов, большая часть которых построена на извлечении ключевой информации из документов [2-4]: TF-IDF, RAKE, YAKE, BERT, TopicRank, TextRank.

В работе для проведения статистического анализа документов (получения его ключевых характеристик) предлагается использовать алгоритм TextRank [5]. Основой данного алгоритма является представление текста в виде графа, где вершинами графа являются слова, с последующим вычислением весов вершин графа.

Связи между вершинами в общем случае формируются исходя из отношений между словами. Обычно используется связь совместного появления слов в пределах определенного окна размера N . Данное отношение является симметричным, и как следствие ребра являются неориентированными. Веса вершин устанавливаются в некоторое начальное значение, равное для всех вершин. Конкретные значения не так важны, т.к. веса относительны. Далее веса итеративно обновляются по формуле:

$$S(V_i) = (1 - d) + d * \sum_{V_j \in \text{In}(V_i)} \frac{w_{ij}}{\sum_{V_k \in \text{Out}(V_j)} w_{jk}} * S(V_j), \quad (1)$$

где V_i - вершина графа; $In(V_i)$ - множество вершин, имеющих связь, направленную в V_i ; $Out(V_i)$ - множество вершин, в которые направлены связи из V_i ; d - коэффициент затухания от 0 до 1 (обычно 0.85); w_{ik} - вес ребра направленного из вершины V_i к вершине V_k .

Обновление весов завершается после определенного кол-ва итераций или по условию, например, как только порядок изменения весов опустится до определенного значения. После вычисления всех весов они упорядочиваются по убыванию. Согласно данному алгоритму, чем выше вес, тем более ключевым является слово для текста.

При реализации алгоритмов анализа документов возникает проблема ресурсоемкости и временных затрат. Можно применить аппаратные решения, однако они будут сильно дорогостоящие и быстро достигнут предела прироста производительности. Перспективным подходом к решению этой проблемы является построение распределенной системы анализа массива документов.

В работе предлагается проектное решение информационно-аналитической системы (ИАС) с организацией распределенной обработки документов на основе брокеров задач (Рисунок 1).

Основными элементами ИАС выступают:

- Gateway – внешний веб-сервер, который доступен конечному пользователю. Основная задача - проксирование запросов на App Instance, а также балансировка нагрузки между ними.
- Database – единое место хранения данных.
- Message Broker – используется для отправки задач от App Instance к Worker Instance.
- App Instance – приложение, обрабатывающее не ресурсоемкие запросы (например, сделать запрос к базе данных и забрать результаты обработки текста, добавить текст в очередь на обработку).
- Worker Instance – приложение, выполняющее ресурсоемкие задачи (например, обработка и анализ текстов).

Оба модуля App Instance и Worker Instance взаимодействуют с брокером сообщений для эффективного распределения задач по обработке документов, а результаты сохраняют в базу данных.

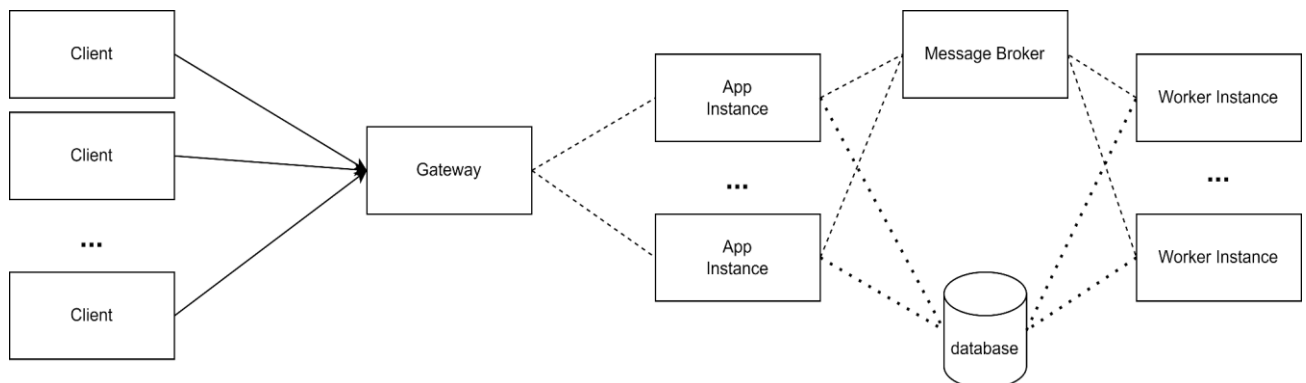


Рисунок 1 – Общая схема распределенной системы.

Для эффективной работы ИАС необходимо учитывать проблему эффективного распределения нагрузки между разными сетевыми ресурсами [6]. В данном случае вычислительными ресурсами выступают приложения по обработке запросов (Instances).

При программной реализации предложенной архитектуры использовались такие технологии:

В качестве Gateway элемента – высокопроизводительный веб- и прокси-сервер Nginx, который может обрабатывать большое количество запросов с минимальной задержкой. Кроме того, он также имеет встроенную балансировку нагрузки, кеширование, защиту от DDoS-атак и многое другое. Модули App Instance реализованы на CPython версии 3.10 с использованием веб-фреймворка FastAPI. Данный фреймворк показывает хорошие показатели на тестах производительности, побеждая по некоторым позициям своих основных конкурентов Flask и Falcon [7].

Для реализации механизма постановки задач в очередь и их выполнения Worker Instance применена технология Celery, общая архитектура обработки запросов в Celery представлена на рисунке 2. В этой архитектуре в качестве брокеров могут выступать Redis и RabbitMQ, а в качестве бэкендов Redis, RabbitMQ, SQLAlchemy (базы данных). В работе использовались и показали эффективность в качестве брокера RabbitMQ и SQLAlchemy как бэкенд для хранения результатов. Так же, как и для App Instance, Worker Instance реализованы на CPython версии 3.10.

В данном решении за работу с данными отвечает мощная, открытая объектно-реляционная система управления базами данных Postgres, которая использует и расширяет язык SQL и имеет множество функций, которые безопасно хранят и масштабируют самые сложные нагрузки по данным.

В ходе масштабирования приложения часто возникают проблемы с ростом соединений к базе данных. Для решения этой проблемы можно использовать легковесный пулер соединений для PostgreSQL – PgBouncer. Он может создавать и повторно использовать соединения к одной или нескольким базам данных (на одном или разных серверах) и обслуживать клиентов по TCP и Unix-

сокетами, поддерживает несколько режимов пула соединений, таких как session, transaction и statement. PgBouncer позволяет уменьшить накладные расходы на открытие и закрытие соединений к PostgreSQL, которые могут быть дорогостоящими в терминах производительности и ресурсов; ограничить количество одновременных соединений; распределять нагрузку между несколькими серверами СУБД; обеспечить высокую доступность СУБД, переключаясь на резервный сервер в случае сбоя основного.

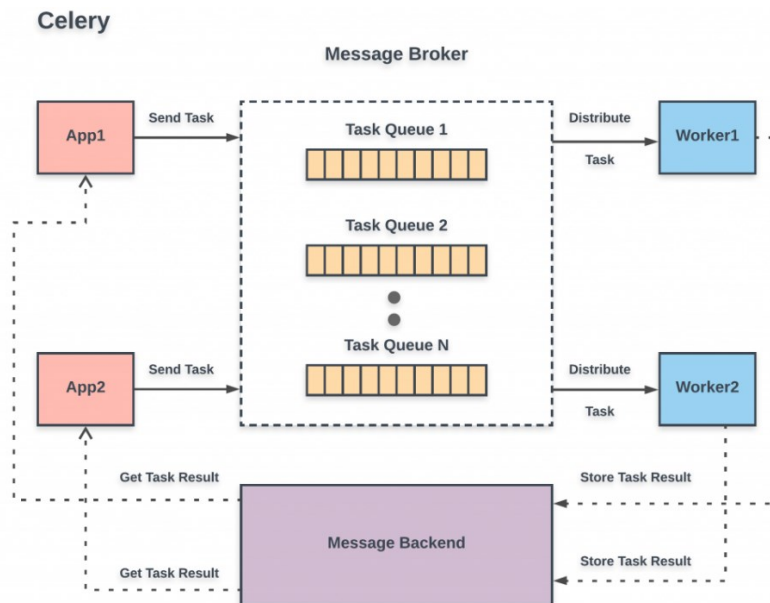


Рисунок 2 – Общая архитектура Celery

Заключение. Таким образом получилась хорошо масштабируемая ИАС, способная достаточно быстро и точно обрабатывать большие массивы документов и классифицировать их по заранее подготовленным категориям. Ожидается, что система позволит повысить эффективность работы с документами за счет сокращения времени поиска.

Список использованных источников:

1. Митина, О.В. Методы анализа текста: методологические основания и программная реализация / О.В. Митина, А.С. Евдокименко // Челябинск: Вестник ЮУрГУ. — 2010. — № 40 (216). — С. 29-38.
2. Методы автоматической классификации текстов [Электронный ресурс]. — Режим доступа: <http://sws.ru/index.php?page=article&id=4252/>. — Дата доступа: 01.04.2023.
3. Keyword Extraction: from TF-IDF to BERT [Электронный ресурс]. — Режим доступа: <https://towardsdatascience.com/keyword-extraction-python-tf-idf-textrank-topicrank-yake-bert-7405d51cd839/>. — Дата доступа: 04.04.2023.
4. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction [Электронный ресурс]. — Режим доступа: https://www.researchgate.net/publication/258908054_TopicRank_Graph-Based_Topic_Ranking_for_Keyphrase_Extraction/. — Дата доступа: 28.03.2023.
5. Милкова, М. А. Современные методы извлечения ключевой информации из нормативных документов / М. А. Милкова, И. В. Неволин, Д. П. Пигорев. // Экономическая наука современной России. — 2021. — № (2). — С. 101-114.
6. Северин, К. М. Программный менеджер распределенных вычислений в мультиагентной среде / К. М. Северин, А. И. Парамонов // Информационные технологии и системы 2022 (ИТС 2022) = Information Technologies and Systems 2022 (ITS 2022) : материалы Международной научной конференции, Минск, 23 ноября 2022 / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. — Минск : БГУИР, 2022. — С. 165–166.
7. Web Framework Benchmarks [Электронный ресурс]. — Режим доступа: <https://www.techempower.com/benchmarks/#section=data-r21&hw=ph&test=composite&a=2&l=zijzen-6bj&c=d&b=4&d=e3>. — Дата доступа: 04.04.2023.

UDC 004.75+004.91

DISTRIBUTED SYSTEM OF STATISTICAL ANALYSIS OF DOCUMENTS

Severin K. M.

Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus

Paramonov A. I. – Candidate of Engineering Sciences, Associate Professor

Annotation. The methods of text analysis are considered. The analysis of the tasks solved by the statistical method is carried out. The problem of automatic classification of text documents is studied, the TextRank algorithm for document classification is considered. The design solution of the information and analytical system of classification of documents is proposed.

Keywords. Document analysis, statistical analysis, TextRank, algorithm, text classification, information and analytical system, graph, distributed system, server, client.

УДК 004.4

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ ПРИЕМНОЙ КОМИССИИ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ

Славинский Г.А.. студент

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Климов С.М. – ст. препод. каф. ИСиТ

Аннотация. Цель проекта – разработка веб-приложения для автоматизации работы приемной комиссии учреждения образования. Проект предназначен для повышения уровня эффективности работы приемной комиссии и улучшения процесса получения, хранения и обработки данных абитуриентов. Веб-приложение ориентировано на потребности учреждения образования «Полоцкий государственный химико-технологический колледж». Результат применим в условиях приемных кампаний учреждений образования, ведущих набор по уровням профессионально-технического и среднего специального образований.

Ключевые слова. Автоматизация работы, приемная комиссия, фреймворк Django.

Введение. В ходе прохождения технологической практики в учреждении образования «Полоцкий государственный химико-технологический колледж», были выделены следующие трудности, возникающие в ходе приемной кампании:

- постоянная необходимость личного присутствия абитуриентов;
- большое число неравномерно поступающих обращений со стороны абитуриентов.

Также, была выделена возможность по улучшению деятельности приемной комиссии через предложение решения для работы с данными абитуриентов, предоставляющего как надежное хранилище, так и гибкий доступ к данному хранилищу.

Приемная кампания является сложным и ограниченным во времени процессом, в ходе которого обрабатываются большие объемы информации [1].

Многие высшие учебные заведения имеют системы для получения, обработки и хранения данных абитуриентов. При поступлении в Белорусский государственный университет информатики и радиоэлектроники каждый абитуриент должен создать личный кабинет, где следует заполнить все необходимые данные и составить заявление с помощью конструктора. Наличие аккаунта позволяет отслеживать статус документов и обращаться в приемную комиссию по возникающим вопросам [2].

Данные системы все более актуальны, независимо от ступени образования, достоинства данного подхода:

- экономия времени, посещение абитуриентом комиссии сводится к подаче оригиналов документов, проверке данных и подписи;
- централизованная система информирования, личный кабинет позволяет абитуриенту следить за ходом вступительной кампании, получить обратную связь от приемной комиссии.

Подобные системы позволяют предварительную подачу документов, но не отменяют необходимость формирования перечня оригинальных документов для подачи в приемную комиссию.

На основе рассмотренных языков (Python, PHP, C#), для разработки был выбран язык программирования Python, как функциональный и более простой в использовании. Из Python-фреймворков был выбран Django, реализующий базовую функциональность веб-приложения на базе шаблона проектирования MVC и содержащий множество функций, позволяющих создать прототип в кратчайшие сроки, включая средства взаимодействия с базой данных и инструменты отладки [3].

Основная часть. При обзоре аналогов программного средства, было рассмотрено как решение для организации данных абитуриентов в рамках файлов СУБД Microsoft Access, так и средство для комплексного управления деятельностью учреждения образования «1С: Колледж», у данных аналогов были выделены следующие достоинства:

- простота, интуитивно понятный интерфейс;
- большое количество материалов для выбранных платформ;
- автоматизация всех этапов деятельности приемной комиссии, включая отчетность.

Выявленные недостатки «1С: Колледж»:

- автоматизация деятельности приемной комиссия не является главной задачей решения;
- возможность выбора нескольких специальностей не актуальна для заказчика.

Выявленные недостатки решений на базе Microsoft Access:

- сложность с организацией доступа к данным;
- отсутствие разграничения доступа.

Главная задача приложения – автоматизация работы приемной комиссии с целью упрощения и качественного улучшения процесса приема абитуриентов. Приложение позволяет легко структурировать и систематизировать данные.

Входные данные предполагают:

— данные абитуриентов – данные, предоставляемые со стороны абитуриента, необходимые для формирования заявления (включают личные данные абитуриентов, данные документов об образовании и представителей абитуриентов).

— организационные данные – данные, предоставляемые со стороны учреждения образования, необходимые для организации процесса приема данных абитуриентов (включают информацию об изучаемых абитуриентами иностранных языках, возможных статусах представителей, актуальных специальностях, установленных при поступлении льготах и т.д.);

Выходные данные предполагают сформированные списки организационных данных, данных абитуриентов, заявлений и документов, предоставленных со стороны абитуриентов.

С целью точной спецификации функций была разработана контекстная диаграмма, представлена на рисунке 1.

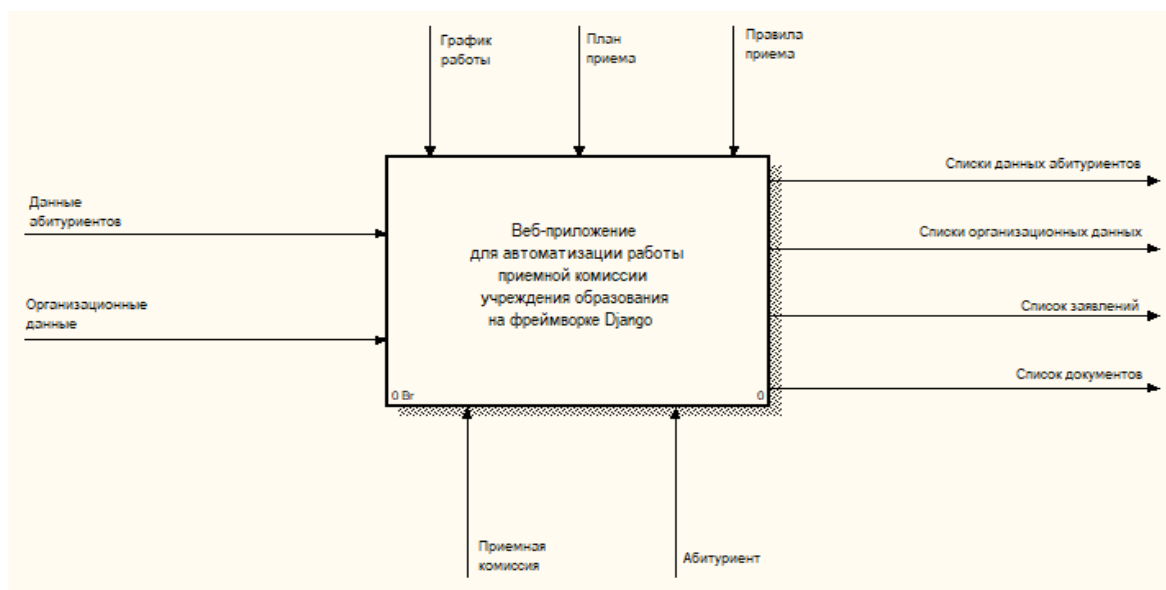


Рисунок 1 – Контекстная диаграмма

Далее была проведена декомпозиция основной конструкции, выполненная согласно принципу ограниченной сложности [4]. Результат декомпозиции представлен на рисунке 2.

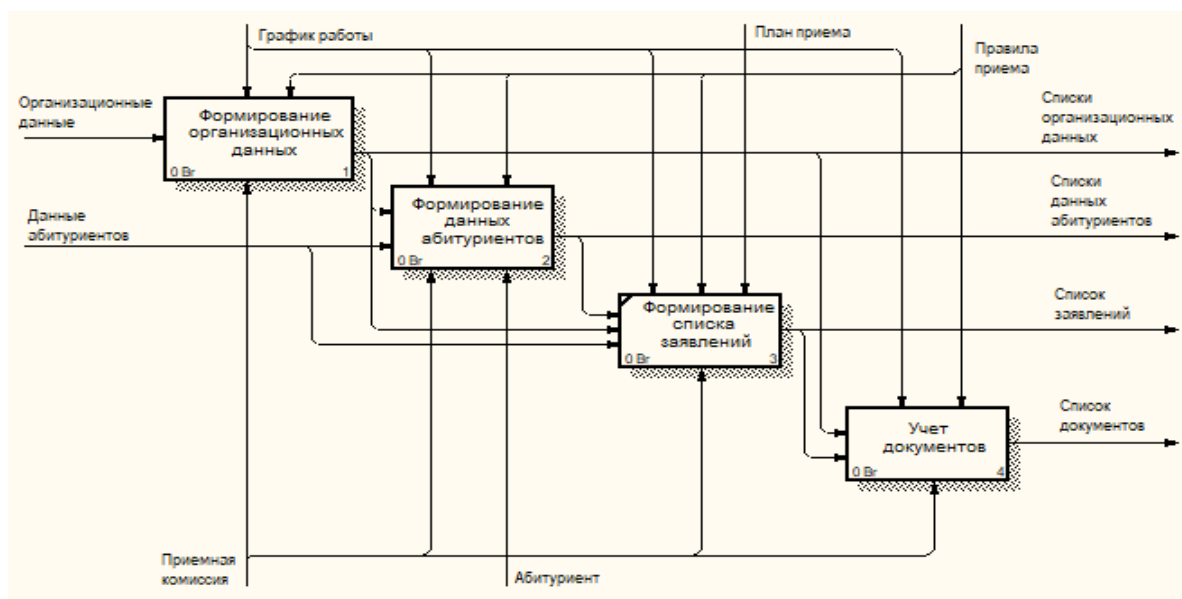


Рисунок 2 – Декомпозиция основной конструкции

Фреймворк Django предлагает готовую таблицу для данных пользователей, содержащую поля логина, пароля и электронной почты. Для реализации механизмов авторизации и регистрации использовались встроенные классы `LoginView`, `AuthenticationForm` и `CreateView`, `UserCreationForm`.

Все категории пользователей, кроме абитуриента, регистрируются через Django-панель.

MVC (Model – View – Controller) – набор архитектурных идей и принципов, сформированных норвежским ученым Трюгве Реенскауг, для построения сложных систем с пользовательским интерфейсом, возможности удобного и лаконичного определения именуется «паттерн» [5].

В фреймворке Django реализуется архитектурный паттерн MTV (Model – Template – View) являющийся модификацией MVC и включающий:

- url-диспетчер – определяет, какой ресурс должен обрабатывать url-запрос;
- model – описывает данные, используемые в приложении;
- template – логика представления, отображает пользователю результат, аналог View в MVC;
- view – получает и обрабатывает запрос, отправляет пользователю некоторый ответ, аналог Controller в MVC.

Функциональные возможности приложения предполагают формирование данных, а точнее – базовые функции работы с БД, реализованные на основе встроенных классов: ListView и DetailView (отображение данных), CreateView (создание), UpdateView (модификация), DeleteView (удаление).

Формирование данных возможно после прохождения авторизации и получения доступа к одной из подсистем, разграничение доступа реализовано через UserPassesTestMixin. Выделенные подсистемы представлены на диаграмме вариантов использования (рис. 3).

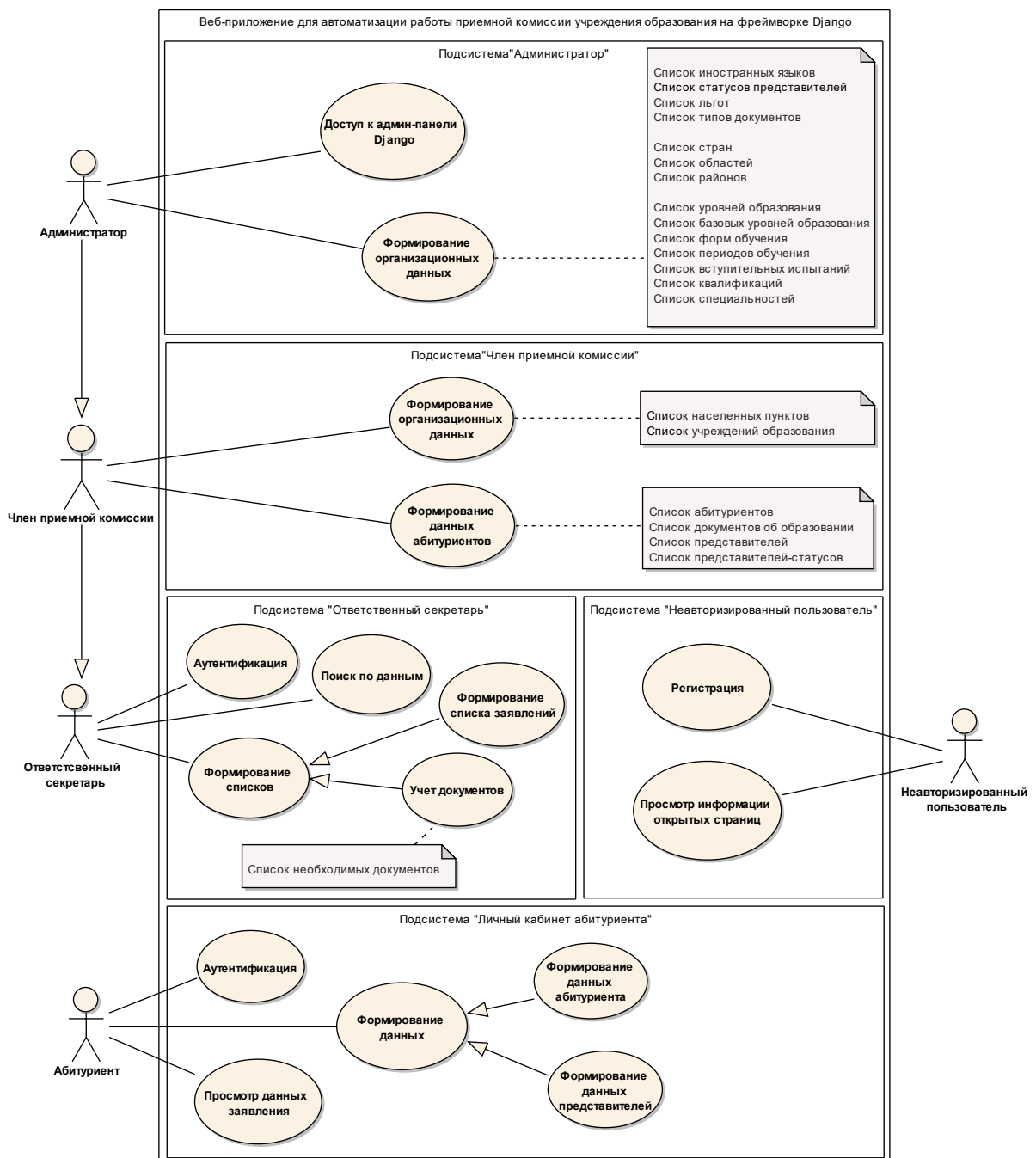


Рисунок 3 – Диаграмма вариантов использования

Перечень сформированных стран представлен на рисунке 4.

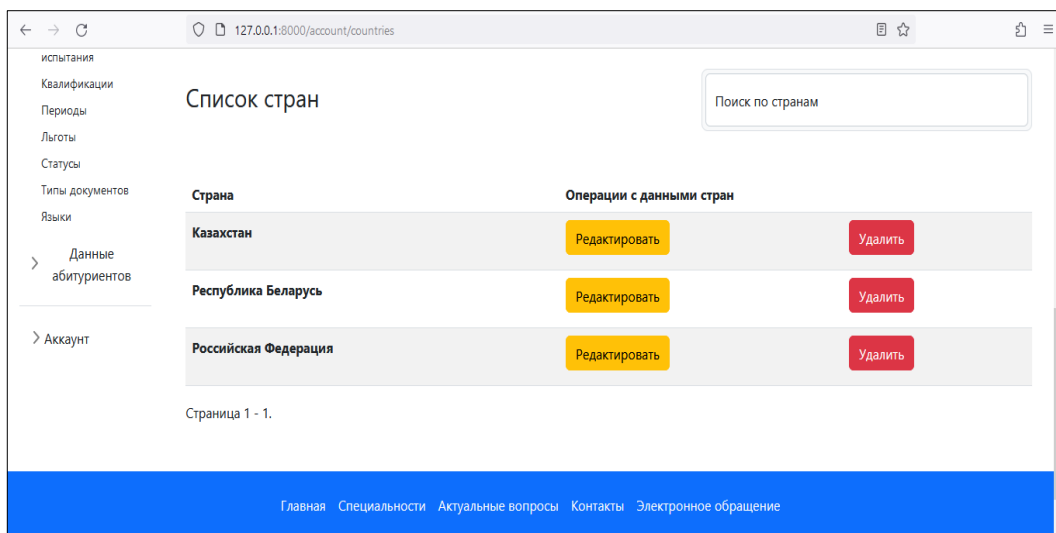


Рисунок 4 – Перечень сформированных стран, подсистема администратора

Добавление данных области, на основе перечня стран, представлено на рисунке 5.

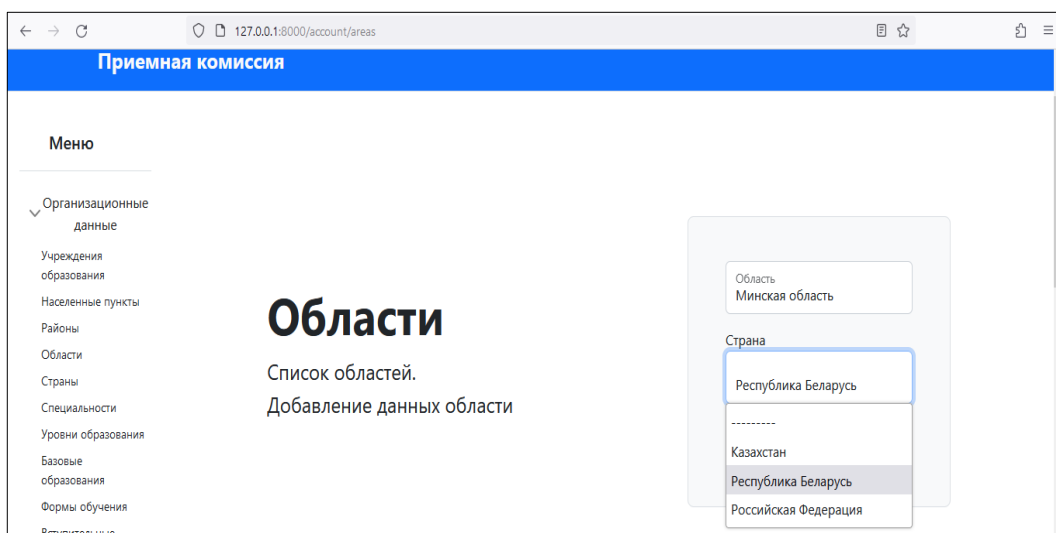


Рисунок 5 – Добавление данных области, подсистема администратора

Результат формирования данных – списки, включая списки абитуриентов, организационных данных, заявлений и поданных документов. Данные заявления представлены на рисунке 6.

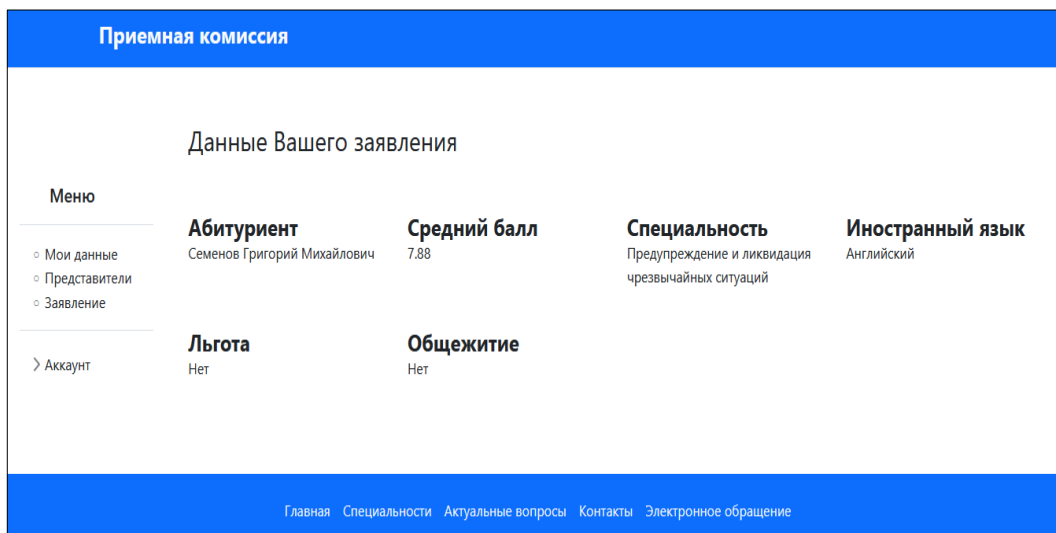


Рисунок 6 – Сформированное заявление, подсистема абитуриента

В случае, если абитуриент еще не обращался в приемную комиссию, то будет представлено уведомление о необходимости посещения приемной комиссии.

Заключение. Веб-приложение реализовано на Python-фреймворке Django с использованием инструмента Bootstrap 5, включающего HTML и CSS шаблоны, и реляционной системы управления базами данных MySQL. Для организации подстановки данных родительских таблиц в поля выбора использовался jQuery-плагин Select2.

В процессе разработки были учтены основные потребности в организации данных абитуриентов на время приемной кампании, реализованы базовые функции, используемые при работе с базами данных: создание, чтение, модификация, удаление.

Приложение отличается от аналогичных программных средств наличием широких функциональных возможностей и своей ориентированностью на автоматизацию работы приемной комиссии.

Технико-экономическое обоснование веб-приложения, в ценах на ноябрь 2022 года за счет средств заказчика, прогнозирует отпускную плановую цену в объеме около 13977,11 рублей.

Проект внедрен в Полоцкий государственный химико-технологический колледж.

Аргументами для внедрения приложения послужило улучшение процесса получения, хранения и обработки данных абитуриентов:

- интуитивно понятный и адаптивный пользовательский интерфейс;
- разграничение прав доступа, согласно выделенным категориям пользователей;
- организация клиент-серверного взаимодействия;
- наличие средств защиты информации от распространенных атак;
- автоматизация рутинных, массово представленных задач, какой, к примеру, является расчет среднего балла документа об образовании абитуриента.

От внедрения ожидается повышение уровня эффективности работы приемной комиссии колледжа. В дальнейшем, с целью оптимизации работы комиссии, предполагается представление приемной комиссии, как многоканальной системы массового обслуживания с неограниченной очередью, рассмотрение возможности внедрения в приложение функционала для оценки эффективности работы сотрудников.

Список использованных источников:

1.1. Батура, М.П. *Повышение эффективности работы приемной комиссии университета путем внедрения автоматизированной системы подачи заявлений и зачисления* / М.П. Батура, Б.В. Никульшин, М.В. Михневич, В.Г. Русин, Н.А. Дубко // *Информационные технологии и системы 2015 (ИТС 2015): материалы международной научной конференции (БГУИР, Минск, Беларусь, 28 октября 2015)* / редкол. : Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2015. – С. 210-211.

2. Приемная кампания-2021 [Электронный ресурс]. – Режим доступа: <https://www.bsuir.by/ru/news/106894-priemnaya-kampaniya-2021-uproschenie-podachi-dokumentov-i-obnovlennye-spetsialnosti-v-bguir>. – Дата доступа: 10.10.2022.

3. Дронов, В. *Django: Практика создания Web-сайтов на Python* / В. Дронов. – СПб : Питер Ком, 2016. – 530 с.

4. Бахтизин, В. В. *Структурный анализ и моделирование в среде CASE-средства VPwin: учеб. пособие по курсу "Технология проектирования программ" для студ. спец. 40 01 01 "Программное обеспечение информационных технологий"* / В. В. Бахтизин, Л. А. Глухова. – Минск: БГУИР, 2002. – 44 с.: ил.

Знакомство с паттерном MVC (Model-View-Controller) [Электронный ресурс]. – Режим доступа: <https://javarush.com/groups/posts/2536-chastjh-7-znakomstvo-s-patternom-mvc-model-view-controller>. – Дата доступа: 07.11.2022

UDC 004.4

WEB APP FOR WORK AUTOMATION OF THE EDUCATIONAL ESTABLISHMENT ADMISSIONS COMMITTEE

Slavinsky G.A.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Klimov S.M. – Senior Lecturer

Annotation. The goal of the project is to develop a web application for work automation of the educational establishment admissions committee. The project is designed to increase the work efficiency level of the admissions committee and improve the process of obtaining, storing and processing applicants' data. The web application is focused on the needs of the Polotsk State College of Chemical Technology. The result is applicable in the admission campaigns of educational establishments which carry out admission on the basis of the levels of professional-technical and secondary specialized education.

Keywords. Work automation, admissions committee, Django framework

УДК 681.5:004.383

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ

Тыманович Н.А., студент, Скудняков Ю.А., канд. техн. наук

*Белорусский государственный университет информатики и радиоэлектроники
Институт информационных технологий
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. Для аппаратного обеспечения микроконтроллерной системы в работе используется технология Bluetooth Low Energy, разработана модульная структура прототипа системы, проведены основные этапы проектирования системы, схемотехнические и конструкторские решения.

Ключевые слова. Аппаратное обеспечение, микроконтроллерная система, модульная структура, этапы проектирования.

Введение. Пользовательский опыт испытывает большие издержки при организации взаимодействия посредством проводных интерфейсов и интерфейсов, имеющих проблемы в организации «горячих» подключений и прочих опций, улучшающих общее качество пользовательского опыта. Однако, начиная появляться стандарты беспроводной связи, которые предусматривают многие трудности и проблемы, которые возникают при создании соответствующих связей, встраиваются в современные микропроцессоры и становятся готовы к проектированию сложных систем на их основе уже сегодня. Одна из таких технологий – Bluetooth Low Energy.

Основная часть. Bluetooth 5 – это наиболее значительное усовершенствование стандарта Bluetooth с момента появления технологии BLE 4.0 (Bluetooth Low Energy в Bluetooth 4.0). Он выводит Bluetooth LE на новый уровень, делает возможными новые варианты использования и улучшает существующие [1,2]. Многопротокольные SoC серии nRF52 обладают высокой производительностью и поддерживают Bluetooth 5. GATT – это аббревиатура от Generic ATTtribute Profile, и он определяет способ, которым два устройства Bluetooth с низким энергопотреблением передают данные туда и обратно, используя концепции, называемые сервисами и характеристиками. Он использует общий протокол данных, называемый протоколом атрибутов (ATT), который используется для хранения сервисов, характеристик и связанных данных в простой таблице поиска с использованием 16-битных идентификаторов для каждой записи в таблице. Сервис может иметь одну или несколько характеристик, и каждый сервис отличается от других сервисов с помощью уникального числового идентификатора, называемого UUID, который может быть либо 16-битным (для официально принятых сервисов BLE), либо 128-битным (для пользовательских сервисов). Характеристики – это основной элемент, с которым предстоит взаимодействовать центральным и периферийным устройством BLE, поэтому важно понимать данную концепцию. Они также используются для отправки данных обратно на периферийное устройство BLE, поскольку также существует возможность записи характеристики.

При проектировании микроконтроллерной системы (МКС) можно использовать однокристалльный микроконтроллер (МК) [3]. В данной разработке используется МК серии nrf5xxx. Он отвечает требованиям сложных приложений, которым требуется параллелизм протоколов, а также богатый и разнообразный набор периферийных устройств и функций. Полностью поддерживается многопротокольность с полным параллелизмом протоколов благодаря поддержке популярных операционных систем реального времени и проектированием продвинутой периферии интерфейсов с DMA, поддерживаются протоколы Bluetooth LE, Bluetooth mesh, Thread, Zigbee, 802.15.4, ANT и собственные стеки 2,4 ГГц. МК спроектирован на базе 32-разрядного ЦП ARM® Cortex™-M4 с модулем с плавающей запятой. Также имеется NFC-A. Имеется множество цифровых периферийных устройств и интерфейсов, таких как высокоскоростной SPI и QSPI для взаимодействия с внешней флэш-памятью и дисплеями, PDM и I2S для цифровых микрофонов и аудио, а также полноскоростное USB-устройство для передачи данных и источника питания для подзарядки аккумулятора. Исключительно низкое энергопотребление достигается за счет сложной встроенной системы адаптивного управления питанием. Общее описание BME280 представляет собой комбинированный цифровой датчик влажности, давления и температуры. BME280 обеспечивает высокие показатели точности влажности, давления и температуры. Датчик поддерживает интерфейсы SPI и I²C и может питаться от 1,71 до 3,6 В для питания датчика V DD и от 1,2 до 3,6 В для питания интерфейса V DDIO. Когда датчик отключен, потребление тока падает до 0,1 мкА. Чтобы адаптировать скорость передачи данных, шум, время отклика и потребляемый ток к потребностям пользователя, могут быть выбраны различные режимы передискретизации, режимы фильтрации и скорости передачи данных. Прототип МКС представляет собой модульную конструкцию, состоящую из 3-х взаимозаменяемых модулей. Каждый модуль может быть заменен на сопоставимый по функционалу. Ниже приведена схема взаимодействия модулей между собой и их назначение (рис. 1).

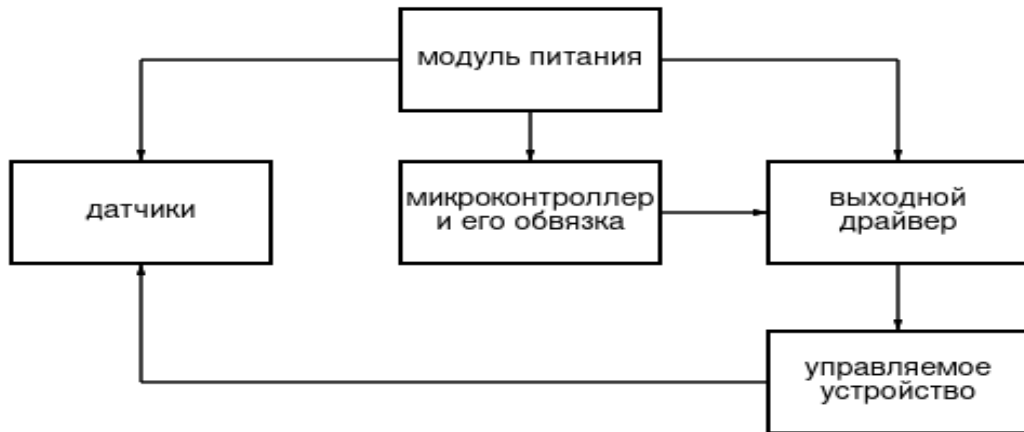


Рисунок 1 – Схема модулей прототипа МКС

Например, конечным управляемым устройством является увлажнитель воздуха.

Существует множество различных принципов работы увлажнителя воздуха.

Рассмотрим только один: пьезоэлемент, который под воздействием постоянно изменяющегося (импульсного или переменного) напряжения начинает деформироваться. Многократно повторяющаяся деформация на высокой частоте заставляет воду, в которой находится пьезоизлучатель, отделяться на достаточно мелкие частицы, чтобы улететь из емкости, образуя таким образом холодный пар. Датчики представляют собой набор отдельных плат (модулей), соединенных по шине SPI или I2C с МК. МК представляет собой отдельную плату, способную взаимодействовать с датчиками по шине SPI/I2C и выходным драйвером через обычные цифровые шины. Может быть организована работа с внешними устройствами, такими как компьютер или телефон для простоты и гибкости управления и настройки через такие интерфейсы, как bluetooth low energy, uart (и его подвид RS-232 и легко преобразуемый в RS-485), USB и т.п. Обвязка будет сильно зависеть от используемых интерфейсов, однако базовый функционал для автономной работы без внешнего взаимодействия фактически ничего не требует. Прототип может разрабатываться на произвольной отладочной плате и/или плате собственного производства. Единственным нюансом является наличие часового и высокочастотного кварца (32 МГц) и, непосредственно, антенны, для Bluetooth Low Energy, а также выхода USB (при его необходимости). Выходной драйвер представляет собой полумостовую схему, обладающую такими показателями, как высокий к.п.д., относительно простая корректировка номинальных режимов заменой комплектующих без изменения схемотехники, относительно высокие нагрузочные токи, работа с индуктивными элементами, такими как трансформаторы и т.п. Схема представлена ниже (рисунок 2).

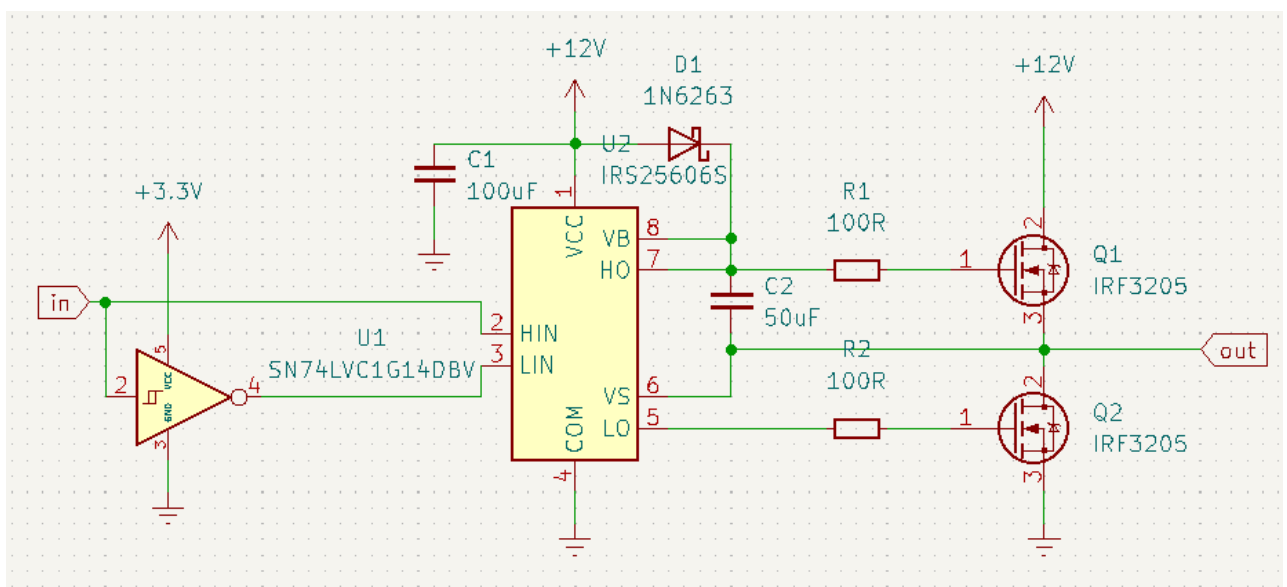


Рисунок 2 – Принципиальная схема драйвера для ШИМ-регулировки ПИД – регулятора

Транзисторы IRF3205 стоят по полумостовой схеме и, в зависимости от требуемых режимов, могут быть заменены на произвольные. Микросхема IRS, конденсаторы, диод и резисторы нужны как

обязательная обвязка и являются основой управления данных транзисторов. Резисторы играют токоограничительную роль, сохраняя работоспособность драйвера от перегрузок. Конденсатор С2 является вольтодобавочным и помогает управлять верхним ключом Q1. Диод заряжает конденсатор С2 и не дает возросшему напряжению на нем при работе верхнего плеча вернуться к питанию микросхемы. Конденсатор С1 просто сглаживает питание. Логический элемент НЕ в микросхеме U1 позволяет управлять драйверами, используя один единственный выход на микроконтроллере.

Таким образом, разработанное устройство, входящее в МКС, обладает рядом преимуществ по сравнению с аналогами: простая и универсальная схема позволяет создать гибкий по цене/возможности драйвер, подходящий для широкого спектра нагрузок. Отсутствие внешних через ПК кнопок/дисплеев удешевляет конструкцию, а возможность вывода и управления непосредственно или смартфоны позволяет обеспечить принципиально новый пользовательский опыт при работе с данным устройством. Современный стек технологий и модульная архитектура позволяют создать устройство, оптимально подходящее под конкретные предложенные технологии нужды по соответствующей цене.

Таким образом, данное устройство и архитектура разработки качественно влияют на конечный продукт и сферы его применения. Разработана компактная и удобная в эксплуатации конструкция МКС, состоящая из 3-х модулей, которая может быть использована для решения задач мониторинга и управления объектами и процессами различного назначения.

Заключение. В процессе проведенного исследования:

- осуществлен анализ возможностей современных программно-аппаратных технологий для создания и эффективного функционирования МКС и входящих в нее беспроводных средств связи;
- осуществлено обоснование выбора МК и датчика;
- разработана схема модулей прототипа МКС, спроектирована принципиальная схема драйвера для ШИМ-регулировки ПИД-регулятора и создана конструкция МКС.

Разработанная МКС предназначена для мониторинга и управления объектами и процессами различного назначения (мобильными и стационарными объектами, производственно-технологическими процессами и т.д.). Разработка МКС прошла все этапы проектирования: начиная от постановки задачи, структурного, функционального, схематехнического и заканчивая конструкторским проектированием.

Для автоматизации функционирования МКС разработано ПО на языках C и Rust.

Использование разработанной МКС обеспечивает:

- повышение производительности ее работы и повышение комфорта при ее эксплуатации;
- гибкость в настройке;
- использование современных программно-аппаратных технологий;
- доступность для расширения функционала;
- минимизацию финансовых издержек при проектировании и изготовлении МКС.

Список использованных источников:

1. Калачев, А. В. Основы работы с технологией Bluetooth Low Energy / А. В. Калачев, М. В. Лапин, М. Е. Пелихов. – Питер : Изд-во Лань, 2020. – 224 с.
2. www.nordicsemi.com [Электронный ресурс] – Режим доступа: <https://www.nordicsemi.com/Products/Bluetooth-Low-Energy/Bluetooth-5>. – Дата доступа: 11.04.2023.
3. Иоффе, В. Г. Структурная организация однокристалльных микроконтроллеров [Электронный ресурс]: учебное пособие / В. Г. Иоффе. – Самара: Изд-во Самарского университета, 2017. – 206 с: ил.

UDC 681.5:004.383

HARDWARE OF THE MICROCONTROLLER SYSTEM

Tymanovich N.A., Skudnyakov Yu.A.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Skudnyakov Yu.A. – Candidate of Engineering Sciences, Associate Professor

Annotation. Bluetooth Low Energy technology is used for the hardware of the microcontroller system, the modular structure of the prototype system has been developed, the main stages of system design, circuit engineering and design solutions have been carried out.

Keywords. hardware, microcontroller system, modular structure, design stages.

УДК 681.32

МОДЕЛИ ОПРЕДЕЛЕНИЯ РАБОТОСПОСОБНОСТИ МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ

Тыманович Н.А., студент, Скудняков Ю.А., канд. техн. наук

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

Скудняков Ю.А. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В работе для обеспечения работоспособности микроконтроллерной системы предложены математические модели с построением траекторий корней на комплексной плоскости относительно окружности единичного радиуса с центром в начале координат, при этом, система будет работать надежно, если корневые траектории располагаются внутри вышеуказанной окружности.

Ключевые слова: микроконтроллерная система, работоспособность, математические модели, траектории корней, комплексная плоскость, окружность единичного радиуса.

Введение. Современные микроконтроллерные системы (МКС) имеют широкое применение в различных сферах человеческой деятельности, в том числе для мониторинга и управления разнопрофильными процессами и объектами. За последнее время архитектура МКС развивалась достаточно быстро [1,2]. Например, МКС интенсивно разрабатываются и используются для мобильных устройств и систем коммуникаций. Разработка и реализация современного математического обеспечения с помощью использования программно-аппаратных средств позволяют МКС достаточно гибко, надежно, с высокой производительностью осуществлять управление различными процессами и объектами искусственного и естественного происхождения. Под мониторингом понимают контроль текущего состояния объекта или процесса, а под управлением – воздействие на объект или процесс с целью обеспечения требуемого их состояния [3], при этом регулирование рассматривается как частный вариант управления для стабилизации выходной величины по значению заданной входной [4].

Основная часть. Целью данной работы является исследование, разработка и практическое применение математических моделей для обеспечения работоспособности МКС в диапазоне ее варьируемых параметров.

Для достижения поставленной цели в работе решаются следующие задачи: 1) анализ существующих разработок; 2) разработка математических моделей, легко поддающихся автоматизации и обладающих наглядной интерпретацией для доступного понимания их возможностей для практического применения; 3) проведение анализа полученных результатов с точки зрения их достоинств, областей применения и перспективы развития. Структура МКС состоит из микропроцессора и ряда других узлов для обработки входной и выходной информации и формирования сигналов управления различными процессами и объектами (рис. 1).

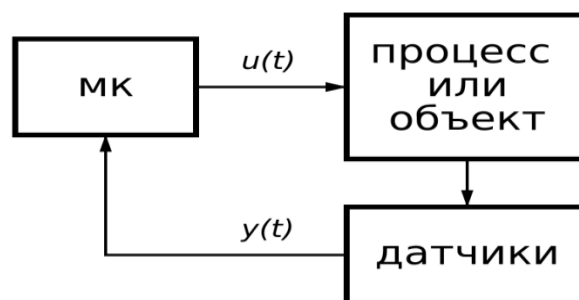


Рисунок 1 – Обобщенная структурная схема МКС

На рисунке 1 обозначены: $u(t)$ – управляющий сигнал; $y(t)$ – сигнал обратной связи (от датчиков); МК – микроконтроллер.

С помощью использования МКС осуществляется мониторинг состояния того или иного процесса и объекта и при отклонении каких-либо (или всех) реальных текущих значений их характеристик и параметров производится приведение их к заданным величинам путем формирования управляющего сигнала для автоматического управления процессов и объектов различной природы и назначения.

Ядром МКС является микропроцессор (или компьютер), обрабатывающий и хранящий данные, необходимые для функционирования системы в целом.

Поскольку от датчиков, кроме цифровых сигналов, могут поступать аналоговые, то на входе компьютера необходимо наличие аналого-цифрового преобразователя (АЦП). Выходной сигнал компьютера имеет цифровую форму и, следовательно, в случае формирования аналогового сигнала необходимо использовать цифро-аналоговый преобразователь (ЦАП).

Все реальные технические средства по своей природе являются нелинейными и, следовательно, МКС в совокупности с другими техническими модулями можно рассматривать как нелинейную цифровую систему управления (НЦСУ).

Работоспособность НЦСУ в значительной степени зависит от стабильной и надежной работы микропроцессора или компьютера и других технических средств. Основным критерием работоспособности НЦСУ является достижение ее абсолютной устойчивости, суть которой заключается в том, что все траектории корней, полученных при решении алгебраических уравнений, описывающих динамику системы, должны находиться внутри окружности единичного радиуса комплексной плоскости $z = \delta^* + j\omega^*$ [5].

НЦСУ состоит из нелинейного элемента (НЭ) с характеристикой $\varphi(\sigma)$, линейной цифровой части (ЛЦЧ), представляющей собой совокупность цифрового элемента с периодом переключения T и приведенной непрерывной линейной части (ПНЛЧ), состоящей из экстраполирующего устройства (ЭУ) и непрерывной части.

Передаточная функция ПНЛЧ, как правило, задается от переменной $p = \delta + j\omega$ и представляет собой трансцендентную функцию, что вызывает большие трудности при оперировании с такой функцией и дальнейшем исследовании.

Поэтому необходим переход от трансцендентной функции к рациональной. Для достижения этой цели применим z – преобразование. Это объясняется тем, что z – преобразование, являющееся преобразованием для цифрового сигнала с присущей ему компактностью и обозримостью выкладок, есть рациональная функция от комплексного переменного $z = \delta^* + j\omega^*$ и, как следствие из этого, – построение корневых портретов цифровых систем можно осуществлять с помощью тех же правил, которые справедливы для построения корневых траекторий непрерывных систем.

В общем виде z – преобразование для передаточной функции ЛЦЧ системы:

$$G^*(z) = \frac{\sum \text{вычеты } G(p)}{1 - e^{pT} z^{-1}} \text{ при полюсах } G(p), \quad (1)$$

где $G(p) = \psi_m(p) / \Phi_n(p)$; $p = \delta + j\omega$; $\psi_m(p)$, $\Phi_n(p)$ – полиномы целочисленных степеней m и n соответственно, $G^*(z)$ – z -преобразование для $G(p)$, где $z = e^{(\delta + j\omega)T}$.

Для случая, когда $\Phi_n(p)$ имеет простые корни, приведенное выше выражение для $G^*(z)$ запишем следующим образом:

$$G^*(z) = \left[\sum_{n=1}^N \frac{\psi_m(p_n)}{\Phi'_n(p_n)} \cdot \frac{1}{1 - e^{-T(p-p_n)}} \right] \text{ при } z = e^{pT}, \quad (2)$$

где $p_1, p_2, p_3, \dots, p_N$ – простые корни $\Phi_n(p_n)$, $\Phi'_n(p_n) = d\Phi/dp_{p=p_n}$

Из двух вышеприведенных формул видно, что функция $G^*(z)$ только переменного $z = \delta^* + j\omega^* = e^{pT}$, где T – период дискретности.

Поскольку $G^*(z)$ – функция только переменного $z = \delta^* + j\omega^* = e^{pT}$, то вся окружность единичного радиуса, лежащая в плоскости z , является конформным отображением частотного годографа $G^*(z)$ плоскости G^* на плоскость z при $z = e^{j\omega T}$.

Следовательно, рассматриваемая система будет работоспособной при условии нахождения всех корневых траекторий внутри окружности единичного радиуса плоскости z .

Проведем анализ работоспособности НЦСУ на основе исследования ее абсолютной устойчивости с нелинейным элементом, стационарная характеристика которого удовлетворяет требуемым условиям и приведенной непрерывной линейной частью с передаточной функцией:

$$G(p) = \frac{1 - e^{-pT}}{p(p+1)}, \quad (3)$$

где $\frac{1 - e^{-pT}}{p}$ – передаточная функция экстраполятора нулевого порядка; $T = 1$ сек. – период дискретности. Поскольку $\Phi_n(p) = p(p+1)$ имеет простые корни $p_1 = 0$, $p_2 = -1$, то получаем:

$$G^*(z) = \frac{0.632}{z - 0.368}. \quad (4)$$

Уравнение корневых траекторий имеет вид:

$$0,632\delta^*k - 0,233k + \delta^{*2} - 0,736\delta^* + \omega^{*2} + 0,135 = 0. \quad (5)$$

Система при $k = 1; 2; 16; 6; 20$ имеет тривиальные корневые траектории, представляющие собой геометрическое место (1) в виде окружностей на плоскости Z (рисунок 2).

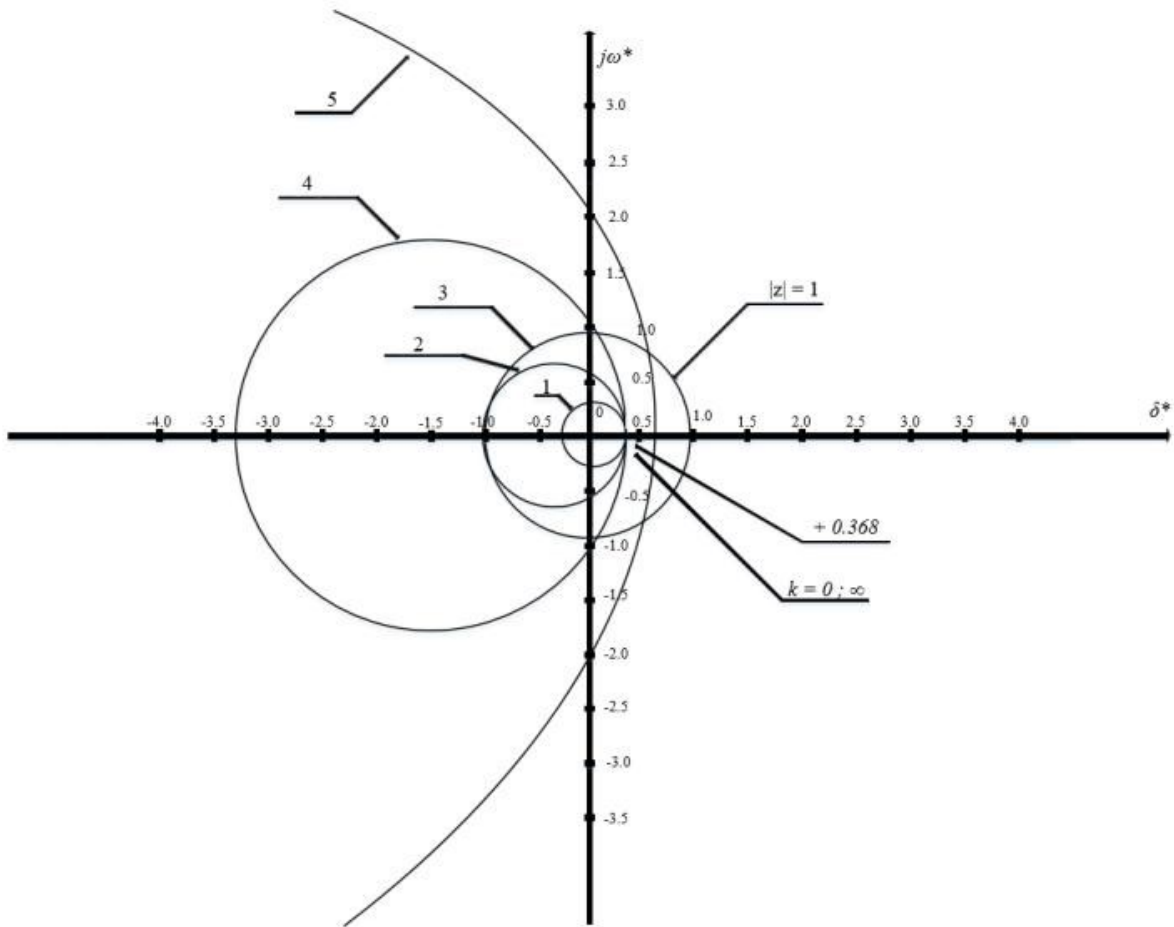


Рисунок 2 – Траектории корней на комплексной плоскости при различных значениях параметров МКС

На рисунке 2 цифрами 1, 2, 4, 5 указаны корневые годографы НЦСУ для $k = 1; 2; 16; 6; 20$ соответственно; цифрой 3 указана окружность единичного радиуса. При увеличении k от 0 до $k_{пр}^*$ НЦСУ становится менее устойчивой поскольку при этом радиус окружностей, увеличиваясь, приближается к единичному значению и равен ему при $k_{пр}^*$. При дальнейшем увеличении k , т.е. при $k > k_{пр}^*$, система – неустойчива, так как в этом случае ветви корневого годографа выходят на внешность окружности единичного радиуса.

В общем случае критической точкой, для которой имеет место $k_{пр}^*$, является любая точка, находящаяся на окружности единичного радиуса, когда радиус – вектор длиной, равной единице, и выходящий из начала координат плоскости Z , описывает эту окружность. Для данной системы такой критической точкой является точка $\delta^* = -1$, при которой $k_{пр}^* = 2.16$.

Следовательно, для обеспечения абсолютной устойчивости рассматриваемой НЦСУ, характеристика $\varphi(b)$ нелинейного элемента должна лежать в угле $[0, k_{пр}^*]$ при заданных параметрах ЛЦЧ.

Для определения работоспособности МКС, содержащих различные нелинейные характеристики, можно строить соответствующие траектории корней на комплексной плоскости Z с использованием следующих формул:

$$\Psi_m^*(Z) = b_0 Z^m + b_1 Z^{m-1} + \dots + b_m; \Phi_n^*(Z) = a_0 Z^n + a_1 Z^{n-1} + \dots + a_n; \quad (6)$$

$$P_m^* = Re\Psi_m^*(Z), R_m^* = Im\Psi_m^*(Z); E_n^* = Re\Phi_n^*(Z), F_n^* = Im\Phi_n^*(Z); \quad (7)$$

$$P_m^* = \sum_{\mu=0}^M \sum_{j=0}^{m-2\mu} (-1)^\mu \omega^{*2\mu} C_{m-j}^{2\mu} b_j \delta^{*m-2\mu-j}; \quad (8)$$

$$R_m^* = \omega^* \sum_{\mu=0}^M \sum_{j=0}^{m-2\mu-1} (-1)^\mu \omega^{*2\mu} C_{m-j}^{2\mu+1} b_j \delta^{*m-2\mu-j-1}; \quad (9)$$

$$E_n^* = \sum_{\mu=0}^N \sum_{j=0}^{n-2\mu} (-1)^\mu \omega^{*2\mu} C_{n-j}^{2\mu} a_j \delta^{*n-2\mu-j}; \quad (10)$$

$$F_n^* = \omega^* \sum_{\mu=0}^N \sum_{j=0}^{n-2\mu-1} (-1)^\mu \omega^{*2\mu} C_{n-j}^{2\mu+1} a_j \delta^{*n-2\mu-j-1}; \quad (11)$$

где $\Psi_m^*(Z)$, $\Phi_n^*(Z)$ – целые полиномы степеней m и n соответственно ($m \leq n$); C_y^λ – биномиальные коэффициенты; $M = \{\frac{m}{2}$ для m четных; $\frac{m-1}{2}$ для m нечетных; $N = \{\frac{n}{2}$ для n четных; $\frac{n-1}{2}$ для n нечетных.

Заключение. Для обеспечения работоспособности спроектированной МКС в данной работе разработаны математические модели определения области ее параметров, в рамках которой система надежно будет выполнять заданные функции. Для достижения этой цели строятся траектории корней на комплексной плоскости, получаемых путем решения алгебраических уравнений, относительно окружности единичного радиуса: если корни лежат внутри данной окружности, МКС – работоспособна для выполнения заданных функций, в противном случае – нет и, следовательно, надо подбирать такое множество параметров системы, при которых она будет соответствовать своему назначению. Разработка МКС прошла все этапы проектирования: начиная от постановки задачи, структурного, функционального, схемотехнического и заканчивая конструкторским проектированием.

Список использованных источников:

1. Иоффе, В. Г. Структурная организация однокристалльных микроконтроллеров [Электронный ресурс]: учебное пособие / В. Г. Иоффе. – Самара.: Изд-во Самарского университета, 2017. – 206 с: ил.
2. Евстифеев, А.В. Микроконтроллеры AVR семейства Мега. Руководство пользователя / А.В. Евстифеев. – М.: Издательский дом «Додэка-XXI», 2007. – 592 с.
3. Сидоренко, Ю.А. Теория автоматического управления: учеб. пособие / Ю.А. Сидоренко. – Минск: БГАТУ, 2007– 124 с.
4. Ковалёв, Д.А. Теория автоматического управления: учебное пособие / Д.А. Ковалёв, В.А. Шаряков, О.Л. Шарякова. – ВШТЭ СПбГУПТД. – СПб., 2020. – 79 с.
5. Скудняков, Ю.А. Модели микроконтроллерной системы для мониторинга и управления объектами и процессами с использованием средств связи / Ю.А. Скудняков, Н.А. Тыманович // Материалы XXVII Международной научно-технической конференции «Современные средства связи», Минск, 27-28 октября 2022 года. – Минск: БГАС, 2022.– С.333-335.

UDC 681.32

MODELS FOR DETERMINING THE OPERABILITY OF A MICROCONTROLLER SYSTEM

Tymanovich N.A., Skudnyakov Yu.A

Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Skudnyakov Yu.A. – Candidate of Engineering Sciences, Associate Professor

Annotation. In order to ensure the operability of the microcontroller system, mathematical models are proposed with the construction of root trajectories on a complex plane relative to a circle of a unit radius centered at the origin, while the system will work reliably if the root trajectories are located inside the above circle.

Keywords: microcontroller system, operability, mathematical models, root trajectories, complex plane, circle of unit radius.

УДК 62-5(075)

МОДЕЛИ СИСТЕМ АВТОМАТИЧЕСКОГО РЕГУЛИРОВАНИЯ ПАРАМЕТРОВ ОБЪЕКТОВ И ПРОЦЕССОВ

Тыманович Н.А., студент, Скудняков Ю.А., канд. техн. наук

*Белорусский государственный университет информатики и радиоэлектроники
Институт информационных технологий
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В работе проведен анализ возможностей и областей практического применения существующих систем автоматического регулирования, приведены их достоинства и недостатки. Рассмотрены основные законы регулирования, реализуемые пропорционально-интегрально-дифференциальными регуляторами, представлены обобщенная структура системы управления с регулятором и ее математические модели. Приведена математическая модель, отражающая полноценный процесс регулирования.

Ключевые слова: автоматическое регулирование, модели, пропорционально-интегрально-дифференциальный регулятор, параметры, объекты, процессы.

Введение. Современные системы автоматического регулирования (САР) имеют широкое применение в различных сферах человеческой деятельности. Например, САР интенсивно разрабатываются и используются для мобильных устройств и систем коммуникаций.

Использование новых структурных решений и математического обеспечения САР позволяет достаточно гибко, надежно, с высокой производительностью осуществлять управление различными процессами и объектами искусственного и естественного происхождения.

Основная часть. Целью данной работы является исследование возможностей, достоинств и недостатков существующих САР и формулировка рекомендаций для практического применения нового структурного решения и эффективного математического обеспечения проектирования САР для стабилизации и управления различного рода процессами и объектами на основе использования пропорционально-интегрально-дифференциального регулятора (ПИД – регулятора) [1-7].

Для достижения сформулированной в работе цели определены следующие задачи исследования:

- 1) анализ существующих разработок;
- 2) постановка решаемой задачи;
- 3) разработка математического обеспечения ПИД – регуляторов;
- 4) обеспечение возможностей для дальнейшей модернизации САР с использованием ПИД – регуляторов;
- 5) проведение анализа полученных результатов.

Практическое использование усовершенствованных ПИД-регуляторов позволяет обеспечивать стабилизацию различного рода параметров среды, таких, например, как влажность, температура, давление и т.д.

Регуляторы в подавляющем большинстве работают по принципу отрицательной обратной связи с целью компенсировать внешние возмущения, действующие на объект или процесс регулирования, и отработать заданный извне или заложенный в системе закон управления. Для определения закона управления используется информация о математической модели объекта или процесса, которая считается заранее известной. В данной работе рассматриваются возможности существующих и перспективных регуляторов.

В рассматриваемой работе проектирование САР осуществляется для организации процесса регулирования на основе пропорционального, дифференциального и интегрального законов управления с использованием ПИД-регуляторов.

Регуляторами обычно называют устройства, которые следят за состоянием объекта или процесса управления и вырабатывают для этого объекта или процесса управляющие сигналы. Регуляторы следят за изменением параметров объекта и процесса управления через, как правило, обратную связь, и реагируют на их изменение с помощью воздействий в соответствии с заданным качеством управления.

Рассмотрим назначение, достоинства и недостатки некоторых существующих ПИД-регуляторов.

ПИД-регулятор ОВЕН ТРМ101 предназначен для регулирования температуры или других физических величин в различных технологических процессах, например, для использования в упаковочном оборудовании. Используется для: экструдеров, термопластавтоматов, печей, упаковочного, полиграфического, вакуумно-формовочного оборудования.

Особенности: измерение температуры или других физических величин; регулирование по ПИД-закону; автонастройка ПИД-регулятора для конкретных объектов регулирования; подключение широкого спектра первичных преобразователей к универсальному входу; дистанционное управление режимами работы прибора; обнаружение обрыва контура (LBA); работа в сети, организованной по стандарту RS-485; регистрация измеренной величины на аналоговом выходе (ток 4...20 мА).

К основным недостаткам системы относятся: ограниченный набор используемых датчиков; отсутствие беспроводных интерфейсов; высокая цена.

Терморегулятор Ringder RC-113M. Особенности: ПИД-регулирование; безрелейное управление; в диапазоне 25 ~ 42С ° точность поддерживаемой температуры может достигать 0,1 С°; широко применяется для террариумов, ремонта инкубаторов, поддержания микроклимата в гроу-боксах и т.д. Параметры: диапазон управления: 0 ~ 60 С°; диапазон измерения температуры: - 40 ~ 110 С°; нагрузка: ~220V / 2A (линейная резистивная нагрузка); электропитание: ~220V; длина провода термодатчика: 1м.; точность может достигать ± 0,1 С° (в диапазоне 25 ~ 42 С°); ± 0,5 С° (другие диапазоны); разрешение: 0.1 С°.

К основным недостаткам системы относятся: малые выходные параметры; отсутствие беспроводных интерфейсов; невозможность подключения средств с нитями накала; относительно высокая цена; отсутствие гибкости и вариативности входных датчиков.

Регулятор температуры – термодат-18Е6, предназначен для использования в различных областях промышленности и производства для автоматизации процессов нагрева, охлаждения и др.

Прибор может работать как обычный ПИД-регулятор или обеспечивать плавное или ступенчатое изменение температуры по программе.

Программа может содержать участки роста/снижения температуры с нужной скоростью и выдержки при заданной температуре.

Назначение каждого выхода задает оператор при настройке прибора. На разных выходах могут быть заданы различные функции.

Результаты измерений с привязкой к реальному времени и дате записываются в энергонезависимую память большого объема, образуя архив данных.

Данные из архива могут быть просмотрены на дисплее прибора в виде графика, переданы на компьютер для дальнейшей обработки или сохранены на USB-носителе.

Подключение к компьютеру осуществляется по интерфейсу RS485. К компьютеру одновременно может быть подключено несколько приборов. Их количество зависит от структуры сети и от используемого на компьютере программного обеспечения.

Прибор поддерживает два протокола обмена с компьютером: «Термодат» - протокол, специфический для приборов «Термодат», и широко распространенный протокол Modbus.

Ключевые особенности: регулирование по программе; электронный самописец; большой графический дисплей 6 дюймов; ПИД-закон регулирования, автоматическая настройка коэффициентов; универсальный вход; логический (цифровой) вход; выходы: релейный, симисторный, транзисторный, аналоговый; USB-разъем; аварийная сигнализация; прочный металлический корпус.

Основные недостатки:

- 1) относительно высокая стоимость системы;
- 2) отсутствие беспроводного интерфейса;
- 3) отсутствие модульной архитектуры.

На основе проведенного обзора наиболее широко применяемых ПИД-регуляторов следует, что актуальной задачей является разработка и практическое применение регуляторов, обладающих высокими надежностью, производительностью, доступной ценой, которыми не в полной мере обладают приведенные выше существующие САР.

Рассмотрим структурную и математические модели систем регулирования.

При отклонении каких-либо (или всех) реальных текущих значений параметров объектов и процессов осуществляется приведение их к заданным величинам путем формирования управляющего сигнала ПИД-регулятором, выполняющим функции автоматического регулирования объектов и процессов различной природы и назначения.

ПИД-регулятор выполняет функцию обратной связи и используется для достижения высоких требований к качеству и точности переходных процессов.

Если какой-то из этих трех компонентов не включен в процесс сложения, то регулятор будет уже не ПИД, а просто пропорциональным, пропорционально-дифференциальным или пропорционально-интегральным.

Схема системы автоматического управления с ПИД-регулятором представлена на рисунке 1.

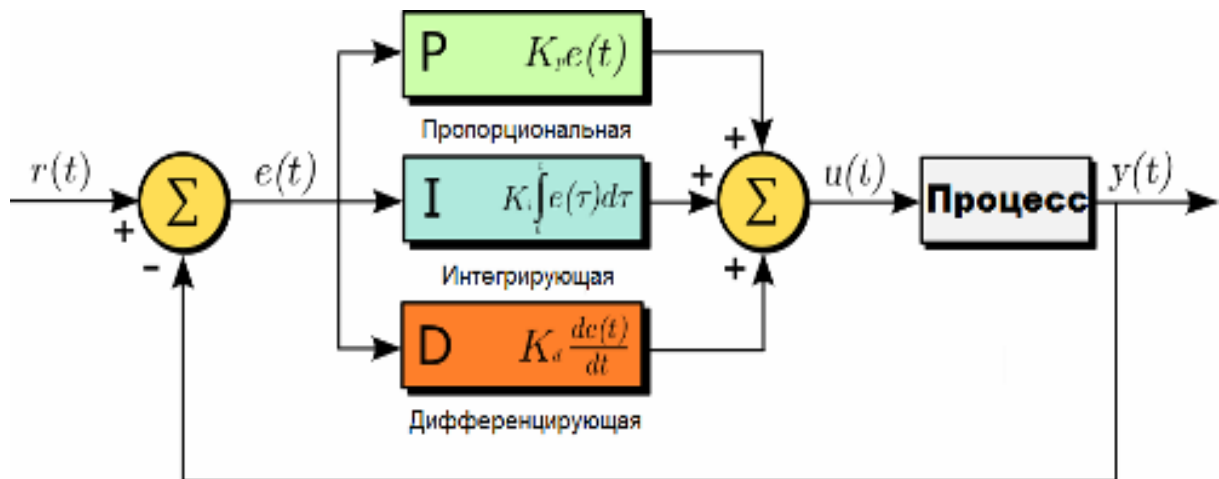


Рисунок 1 – Схема системы автоматического управления с ПИД-регулятором

На рисунке 1 обозначены: $r(t)$ – задающий сигнал; $e(t) = r(t) - y(t)$ – разность между задающим сигналом и сигналом обратной связи; формулы, описывающие пропорциональный, интегральный и дифференциальный законы регулирования соответственно:

$$P = K_p e(t), \quad I = K_i \int_0^t e(\tau) d\tau, \quad D = K_d \frac{de(t)}{dt}, \quad (1)$$

где K_p , K_i , K_d – коэффициенты передачи (усиления) регулятора; $u(t)$ – управляющий сигнал; $y(t)$ – сигнал обратной связи. Использование интегрального закона I позволяет учитывать все множество возмущений, отрицательно влияющих на состояние процесса или объекта, и регулировать выходную величину $y(t)$ по отклонению. В этом случае управление является более точным по сравнению с управлением по возмущению, но осуществляется с определенным временным запаздыванием τ .

При использовании пропорционального закона регулирования выходной сигнал приводит к противодействию текущему отклонению входной величины. Чем больше отклонение – тем больше сигнал. Когда на входе значение регулируемой величины равно заданному, то выходной сигнал становится равным нулю.

Если использовать только пропорциональную составляющую, то значение величины, подлежащей регулированию, никогда не стабилизируется на правильном значении. Всегда есть статическая ошибка, равная такому значению отклонения регулируемой величины, что выходной сигнал стабилизируется на этом значении.

К примеру, терморегулятор управляет мощностью нагревательного прибора. Выходной сигнал уменьшается по мере приближения требуемой температуры объекта, и сигнал управления стабилизирует мощность на уровне тепловых потерь.

В итоге заданного значения температура так и не достигнет, ибо нагревательный прибор просто должен будет быть выключен, и начнет остывать (мощность равна нулю). Больше коэффициент усиления между входом и выходом – меньше статическая ошибка, но если коэффициент усиления (по сути – коэффициент пропорциональности) будет слишком большим, то при условии наличия задержек в системе (а они зачастую неизбежны), в ней вскоре начнутся автоколебания, а если увеличить коэффициент еще больше – система попросту утратит устойчивость.

Или пример позиционирования двигателя с редуктором.

При малом коэффициенте нужное положение рабочего органа достигается слишком медленно. Увеличить коэффициент – реакция получится более быстрая. Но если увеличивать коэффициент дальше, то двигатель «перелетит» правильную позицию, и система не перейдет быстро к требуемому положению, как хотелось бы ожидать. Если теперь увеличивать коэффициент пропорциональности дальше, то начнутся осцилляции около нужной точки – результат снова не будет достигнут.

Интеграл по времени от величины рассогласования – есть основная часть интегральной составляющей. Она пропорциональна этому интегралу. Интегрирующий компонент используется для исключения статической ошибки, поскольку регулятор со временем учитывает статическую погрешность.

В отсутствие внешних возмущений, через какое-то время подлежащая регулированию величина будет стабилизирована на правильном значении, когда пропорциональная составляющая окажется равной нулю, и точность выхода будет целиком обеспечена интегрирующей составляющей. Но интегрирующая составляющая тоже может породить осцилляции около точки позиционирования, если коэффициент не подобран правильно.

Темпу изменения отклонения величины, подлежащей регулированию, пропорциональна третья – дифференцирующая составляющая. Она необходима для того, чтобы противодействовать

отклонениям (вызванным внешними воздействиями или задержками) от правильного положения, прогнозируемого в будущем. Если система линейна и стационарна (практически это вряд ли возможно), то для реализации задания $u(t)$ справедлива следующая формула:

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}. \quad (2)$$

С практической точки зрения в ПИД-регуляторах для их настройки лучше использовать формулу, где коэффициент передачи (усиления) применен сразу ко всем компонентам [8]:

$$u(t) = K_p(e(t) + K_{ip} \int_0^t e(\tau) d\tau + K_{dp} \frac{de(t)}{dt}). \quad (3)$$

Заключение. Итак, в процессе проведенного исследования:

— рассмотрены основные законы регулирования, реализуемые ПИД-регуляторами для коррекции и стабилизации необходимых характеристик и параметров различных процессов и объектов в случае отклонения значений последних от заданных величин по результатам проведенного мониторинга;

— рассмотрены структурная и математические модели функционирования различных типов регуляторов;

— показано, что полноценное ПИД-регулирование возможно только при совместной реализации всех трех основных законов регулирования;

— программная реализация рассмотренных моделей осуществлена на языке С.

Список использованных источников:

1. Ang, K.H. *PID control system analysis, design, and technology* / K.H. Ang, G. Chong, Y. Li // *IEEE Transactions on Control Systems Technology*. 2005. Vol. 13. No. 4. P. 559-576.
2. O'Dwyer, A. *PID compensation of time delayed processes 1998-2002: a survey* / A. O'Dwyer // *Proceedings of the American Control Conference, Denver, Colorado, 4-6 June 2003*. – P. 1494 – 1499.
3. Quevedo, J. *Digital control: past, present and future of PID control* / J. Quevedo, T. Escobet // *Proceedings of the IFAC Workshop, Eds., Terrassa, Spain, 5-7 Apr. 2000*.
4. Astrom, K.J. *Advanced PID control*. / K.J. Astrom, T. Hagglund // *ISA (The Instrumentation System, and Automation Society)*, 2006. – 460 p.
5. Li, Y. *Patents, software, and hardware for PID control. An overview and analysis of the current art* / Y. Li, K.H. Ang, G.C.Y. Chong // *IEEE Control Systems Magazine*. Feb. 2006. – P. 41-54.
6. Frantsuzova, G. A. *Calculation of robust PID- controller* / G. A. Frantsuzova, N. S. Zemtsov, L. Hubka, O. Modrlak // *Proc. of the 12th Intern. Conf. "Actual Problems of Electronic Instrument Engineering (APEIE-2014)"*. Novosibirsk, 2014. Vol. 1. – P. 675-678.
7. Французова, Г.А. Расчет параметров робастного ПИД-регулятора на основе метода локализации / Г.А. Французова, Н.С. Земцов // *Вести ЮУрГУ. Сер. Компьютерные технологии, управление, радиоэлектроника*. 2013. 13, №4. – С. 134-138.
8. Тыманович, Н.А. *Микроконтроллерная система для мониторинга и управления процессами и объектами различного назначения* / Н.А. Тыманович, Ю.А. Скудняков // *XXI Международная конференция «Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2022)»*, Минск, 17 ноября 2022 г. – Минск : ОИПИ НАН Беларуси, 2022. – С. 232-235.

UDC 62-5(075)

MODELS OF SYSTEMS FOR AUTOMATIC CONTROL OF PARAMETERS OF OBJECTS AND PROCESSES

Tymanovich N.A., Skudnyakov Yu.A.

Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Skudnyakov Yu.A. – Candidate of Engineering Sciences, Associate Professor

Annotation. The paper analyzes the possibilities and areas of practical application of existing automatic control systems, their advantages and disadvantages are given. The basic laws of regulation implemented by proportional-integral-differential regulators are considered, the generalized structure of the control system with a regulator and its mathematical models are presented. A mathematical model reflecting a full-fledged process of regulation is given.

Keywords: automatic regulation, models, proportional-integral-differential regulator, parameters, objects, processes.

УДК 004.031.6

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОКОНТРОЛЛЕРНОЙ СИСТЕМЫ

Тыманович Н.А., студент, Скудняков Ю.А., канд. техн. наук

*Белорусский государственный университет информатики и радиоэлектроники
Институт информационных технологий
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В данной работе для автоматизации функционирования микроконтроллерной системы разработано программное обеспечение, использование которого позволяет повысить производительность и качество обработки информации системой.

Ключевые слова. автоматизация, микроконтроллерная система, производительность, качество, обработка информации.

Введение. Высокопроизводительная, надежная и высокоточная работа современной микроконтроллерной системы (МКС) возможна только при использовании эффективного программного обеспечения (ПО). Создание высококачественной МКС, выполняющей заданные функции в автоматическом режиме, является актуальной задачей. В рамках данной работы разработано ПО для МКС, реализующей функции мониторинга и управления объектами и процессами разного назначения.

Основная часть. Цель работы состоит в исследовании, разработке и практическом применении ПО построения и функционирования МКС для повышения эффективности контроля состояний управляемых объектов и процессов и корреляции их характеристик и параметров на основе использования обратной связи и референсных значений.

При выполнении работы проведен анализ существующих проблем и подходов, имеющих место в области автоматического мониторинга, управления и стабилизации динамических величин, метрик и параметров, которые не могут быть линейно и предсказуемо установлены.

В результате проведения исследований разработано ПО для эффективного мониторинга и управления нелинейных динамических величин, которые нельзя статически задать при изготовлении изделия.

Применение полученных результатов разработки и последних достижений в области автоматизации мониторинга и управления различных объектов и процессов позволяет создавать автономные комплексы, которые могут эффективно сохранять и стабилизировать параметры системы, несмотря на изменяющиеся внешние факторы.

При разработке ПО МКС, с привязкой к аппаратным решениям, используются компиляция и линковка, которые имеют свою специфику при работе с bare metal проектами, учитывая проводные и беспроводные интерфейсы, датчики и микроконтроллеры (МК), которые непосредственно взаимодействуют с ними. За основу взят официальный SDK производителя на базе проекта Zephyr. Zephyr – это операционная система реального времени встраиваемых систем для МК. Zephyr включает в себя ядро, компоненты и библиотеки драйверов устройств, стеков протоколов, файловых систем и обновления встроенного ПО, необходимые для разработки полноценного прикладного ПО. Использовать SDK производителя вынуждает не только то, что туда, помимо непосредственно, драйверов периферии МК, входят некоторые драйверы датчиков, но и то, что актуальная реализация bluetooth low energy стека выполнена именно на базе данной ОСРВ. Но это, в свою очередь, накладывает некоторые ограничения на использование сторонних языков программирования, которые изначально несовместимы с C, такие как, например, Rust.

Чтобы использовать SDK производителя, имея полный доступ ко всем возможностям МК, но не ограничиваться C, его библиотеками и т.п., а можно поступить следующим образом: основа сборки и уровень доступа к периферии, вместе с ее драйверами, будут использоваться из SDK производителя, но логика работы приложения может быть выполнена на любом другом компилируемом языке программирования без сборки «мусора», который может иметь совместимость с C ABI. Достаточно реализовать некоторые логические компоненты на нужном языке программирования, собрать его как статическую библиотеку в C ABI, после чего использовать полученные функции непосредственно в нужных местах проекта на базе SDK, предварительно указав статическую библиотеку в соответствующих файлах конфигурации сборки. Таким образом, можно использовать произвольный инструментарий и почти нет ограничений в наборе технологий, с которыми можно работать. Почему почти? Работа с прерываниями должна проходить только через SDK производителя, если он хоть где-то участвует в проекте. Дело в том, что, если появится необходимость использовать прерывания на другом языке программирования, придется создавать таблицу векторов и оперировать ей, но SDK производителя будет также иметь свою версию таблицы векторов, часть которых он уже реализовал, что, в свою очередь, приведет к ошибке линковки. При удалении из SDK производителя таблицы

векторов не гарантируется работоспособность всех его компонентов, включая BLE стек, планировщик ОСРВ и т.п.

Рассмотрим особенности линковки и сборки прошивки под встраиваемые системы.

Языки программирования представляют собой средство описания вычислений для людей и машин. Прежде чем запустить программу, ее необходимо преобразовать в форму, которая может выполняться на компьютере. Генерация переносимой программы на машинном языке (часто именуемой объектным модулем (object module)) обеспечивает возможность отдельной компиляции подпрограмм. Набор переместимых объектных модулей может быть скомпонован в одно целое и загружен для выполнения [1].

Создание скрипта компоновщика – неотъемлемый процесс при создании экосистемы для новых микропроцессоров [2,3]. Без него не будет возможности выполнения исполняемой прошивки, так как для каждого МК необходимо, в зависимости от архитектуры, предпочтение производителя и т.п., особая подготовка конечного бинарного файла в соответствующую форму.

Процесс компиляции под bare metal (процессоры без установленной OS), в общем, справедлив и для x86/amd64 процессора, за исключением некоторых особенностей: скомпилированная программа выполняется не на том же устройстве, на котором она компилируется, появляются дополнительные опции для указания модели памяти и таблицы векторов прерываний. По этой причине, хоть компоновщик и является частью компилятора, его стоит отдельно рассматривать в контексте, зависящем от таких факторов, как, например, наличие/отсутствие OS, архитектуры, программной модели и т.п.

В большинстве случаев компилятор может сам получить всю необходимую информацию для подготовки конечного исполняемого объекта, но иногда, всё же, приходится вмешиваться в процесс его работы. Хотя принято считать стартовой точкой программы функцию main(), но это, на самом деле, не совсем корректно. До вызова «главной функции» происходит довольно большое количество операций.

Когда запускается сборка проекта, каждый модуль собирается в объектный файл. По большей части он состоит из машинного кода под заданную архитектуру, но он не является «автономным», т.е. не может быть запущен. Компилятор помещает туда дополнительную информацию: ссылки на функции и переменные, определённые вне модуля в виде таблицы.

На рисунке 1 представлена схема этапов, начиная от проектирования ПО и заканчивая вызовом main функции.



Рисунок 1 – Этапы проектирования ПО

Каждый объектный файл состоит из одной или нескольких секций, в которых хранится либо код, либо данные. Для GCC программный код складывается в секцию .text, проинициализированные глобальные переменные с их значениями складываются в секцию .data, обнуленные - в секцию .bss. Выходной файл компоновщика – это такой же объектный файл, с точно таким же форматом хранения кода и данных. Другими словами, компоновщик группирует код и данные по секциям, пытаясь разрешить внешние связи. Такой файл, однако, не может быть исполнен на целевой платформе. Проблема в том, что в нём нет информации об адресах, где данные и код должны храниться. Следующим в игру вступает локатор.

Любая программа, на любом языке, имеет некоторые требования к среде. Для Java это виртуальная машина, для Python – интерпретатор, а для C – наличие памяти для стека (грубое упрощение). Место под стек должно быть выделено до начала выполнения программы, и этим занимается startup-файл. Он же выполняет и другие задачи, например, производит определенные действия с прерываниями, перемещает нужные данные в оперативную память, и только после этого вызывает функцию main (причем, функция может называться по-другому).

В некоторых компиляторах предусмотрена отдельная утилита, которая занимается локацией адресов, т.е. сопоставлению физических адресов в памяти целевой платформы соответствующим секциям. В GCC она является частью компоновщика. Информация, необходимая для данной процедуры, хранится в специальном файле – в скрипте компоновщика. При работе с интегрированной средой разработки данный файл генерируется автоматически, исходя из указанных параметров МК. В

некоторых случаях бывает полезно изменить его, чтобы добиться желаемого результата: например, поместить код в оперативную память и выполнять программу оттуда.

Рассмотрим основные составляющие данного файла. Текстом «ENTRY (Reset Handler)» указывается компоновщику, с чего начинать запуск. Функция «ResetHandler» определена в startup-файле, её смысл пояснен чуть выше, startup-файл берёт значение этой переменной именно отсюда. Затем определяется минимальный размер для кучи и стека. Эти данные используются в самом конце сборки, когда компоновщик проверяет, достаточно ли места в памяти. В скрипте также должны быть перечислены все виды памяти, доступные в системе.

Рассмотрим подробно механизм реализации и обработки прерываний Interrupt and Events. В МК архитектуры Cortex-M есть два понятия, которые часто путают Interrupt и Event. Event – это событие (аппаратное или программное), на которое могут реагировать ядро или периферийные блоки. Одним из вариантов реакции может быть – прерывание. Interrupt – это прерывание работы программы и переход управления в специализированный участок – обработчик прерывания. Взаимосвязь между Event и Interrupt заключается в следующем: каждый Interrupt вызывается Event, но не каждый Event вызывает Interrupt.

Помимо прерываний, события могут активировать и другие возможности МК. Управление и обработка прерываниями производится контроллером приоритетных векторных прерываний NVIC (Nested Vectored Interrupt Controller).

При возникновении некоторого события контроллер прерываний автоматически прерывает выполнение основной программы и вызывает соответствующую функцию обработки прерываний. После выхода из функции обработчика прерываний программа продолжает выполнение с того места, где произошло прерывание. Все происходит автоматически (при правильной настройке NVIC, но об этом ниже). Из самого названия видно, что контроллер NVIC поддерживает вложенность прерываний и приоритеты. Каждому прерыванию при настройке NVIC присваивается свой приоритет. Если во время обработки низкоприоритетного прерывания возникает высокоприоритетное, то оно, в свою очередь, прервет обработчик низкоприоритетного прерывания.

При инициации прерывания NVIC переключает ядро в режим обработки прерывания. После перехода в режим обработки прерывания регистры ядра помещаются в стек. Непосредственно во время записи значения регистров в стек осуществляется выборка начального адреса функции обработки прерывания. В стек перемещается регистр статуса программы (Program Status Register (PSR)), счетчик программы (Program Counter (PC)) и регистр связи (Link Register (LR)). Благодаря этому запоминается состояние, в котором находилось ядро перед переходом в режим обработки прерываний. Также сохраняются регистры R0 – R3 и R12. Эти регистры используются в инструкциях для передачи параметров, поэтому, помещение в стек делает возможным их использование в функции обработки прерывания, а R12 часто выступает в роли рабочего регистра программы.

По завершении обработки прерывания все действия выполняются в обратном порядке: извлекается содержимое стека и, параллельно с этим, осуществляется выборка адреса возврата. С момента инициации прерывания до выполнения первой команды обработчика прерываний проходит 12 тактов, такое же время необходимо для возобновления основной программы после завершения обработки прерывания. NVIC поддерживает прерывания с различными приоритетами, которые могут прерывать друг друга. При этом могут возникнуть различные ситуации, обработка которых по-разному оптимизирована. Например, приостановка низкоприоритетного прерывания. В этой ситуации обработка низкоприоритетного прерывания прекращается. В следующие 12 циклов выполняется сохранение в стек нового набора данных и запускается обработка высокоприоритетного прерывания. После его обработки, содержимое стека автоматически извлекается и возобновляется обработка низкоприоритетного прерывания. Больших отличий от прерывания основной программы не наблюдается.

Также возможен вариант непрерывной обработки прерываний. Эта ситуация может возникнуть в двух случаях: 1) если два прерывания имеют одинаковый приоритет и возникают одновременно, 2) если низкоприоритетное прерывание возникает во время обработки прерывания с более высоким приоритетом, то в этом случае, промежуточные операции над стеком не производятся: происходит только загрузка адреса обработчика низкоприоритетного прерывания и переход к его выполнению. Отказ от операций над стеком экономит 6 тактов. Переход к следующему прерыванию происходит не за 12 тактов, а всего за 6. Иногда происходит запаздывание высокоприоритетных прерываний. Ситуация возникает, если высокоприоритетное прерывание происходит во время перехода к обработке низкоприоритетного (за те самые 12 тактов). В этом случае переход к высокоприоритетному прерыванию будет происходить не менее 6 тактов с момента его возникновения (время, необходимое для загрузки адреса обработчика прерывания и перехода к нему).

Возврат в низкоприоритетное уже описан выше.

Помимо простой установки приоритета прерываний, NVIC реализует возможность группировки приоритетов. Прерывания в группе с более высоким приоритетом могут прерывать обработчики прерываний группы с более низким приоритетом. Прерывания из одной группы, но с разным приоритетом внутри группы, не могут прерывать друг друга. Приоритет внутри группы определяет только порядок вызова обработчика, когда были активизированы оба события.

Для того, чтобы включать/выключать различные векторы прерываний, существует маскирование прерываний. Если прерывание замаскировано, это не означает, что периферия не генерирует события. Просто NVIC не вызывает обработчик этого события. Все возможные прерывания, поддерживаемые NVIC, записываются в таблицу векторов прерываний.

По сути своей, таблица векторов прерываний есть ничто иное как список адресов функций обработчиков прерываний. Номер в списке соответствует номеру прерывания. NVIC поддерживает до 240 различных векторов прерываний. Но реализация уже зависит от конкретного производителя. В описании ядра стандартизованы только прерывания исключений ядра: Reset; NMI; HardFault; MemManage; BusFault; UsageFault; SVC; PendSV; SysTick.

Из начала флеш-памяти ядро считывает значение SP (stack top address) и PC (reset routine location).

Таким образом, автоматически начинается выполняться функция с адресом, считанным в регистр PC. Это может быть, например main. Кроме наличия обязательных четырех компонентов, может находиться дальнейшая таблица векторов прерываний. Можно разместить таблицу векторов прерываний в другой области памяти, но тогда необходимо сообщить NVIC, куда передвинута таблица.

Заключение. В итоге поведенного исследования:

- разработана схема этапов проектирования ПО МКС;
- рассмотрены особенности компилятора для разработки и применения ПО в процессах проектирования и функционирования МКС;
- ПО МКС разработано на языках C [4] и Rust [5,6];
- код на языке C используется для взаимодействия с датчиками и BLE, по которому может осуществляться передача данных. SDK (набор инструментов) производителя микроконтроллера использует C, а стало быть, если потребуется использовать проприетарные протоколы, такие как BLE, придется использовать их SDK на C.

Язык Rust является более мощным и комфортным в использовании, так как он предоставляет более гибкие механизмы при написании кода, такие как, например, замыкания, дженерики (generics) и др. Поэтому было принято решение использовать его для написания на нем ПО ПИД – регулятора. Так как Rust может использовать C ABI, можно отдельно применять оба языка в одном проекте.

Список использованных источников:

1. Ахо, А. Компиляторы. Принципы, технологии и инструментарий / А. Ахо, Д. Ульман, М. Лам. – М.: Изд-во Диалектика, 2019. – 29 с.
2. Руководство новичка по эксплуатации компоновщика [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/150327/>. – Дата доступа: 17.09.2022.
3. Создание скриптов компоновщика [Электронный ресурс]. – Режим доступа: <https://www.tuneit.ru/web/tyaut/home/-/blogs/25692/>. – Дата доступа: 17.09.2022.
4. Демидович, Е.М. Основы алгоритмизации и программирования. Язык СИ / Е.М. Демидович. – Минск: Бест-принт, 2001. – 440 с.
5. Клабник, С. Программирование на Rust / С. Клабник, К. Николс. – Питер: ЛитРес, 2021. – 592 с.
6. Тыманович, Н.А. Программирование на языке Rust в Embedded-системах / Н.А. Тыманович, Ю.А. Скудняков // Материалы XXV Международной научно-технической конференции «Современные средства связи», Минск, 22-23 октября 2022 года. – Минск: БГАС, 2022. – С. 174.

UDC 004.031.6

MICROCONTROLLER SYSTEM SOFTWARE

Tymanovich N.A., Skudnyakov Yu.A.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Skudnyakov Yu.A. – Candidate of Engineering Sciences, Associate Professor

Annotation. In this paper, software has been developed to automate the functioning of the microcontroller system, the use of which allows to increase the productivity and quality of information processing by the system.

Keywords. Automation, microcontroller system, productivity, quality, information processing.

УДК 004.42

ВЕБ-ПРИЛОЖЕНИЕ «ВЫПУСКАЮЩАЯ КАФЕДРА» ДЛЯ АВТОМАТИЗАЦИИ И ЦИФРОВИЗАЦИИ ДИПЛОМНОГО ПРОЕКТИРОВАНИЯ

Хотак С.А., студент, Савенко А.Г., маг. техн. наук

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Савенко А.Г. – маг. техн. наук, ст. препод. каф. ИСиТ

Аннотация. В данной работе представлено разработанное веб-приложение, предназначенное для автоматизации и цифровизации процесса дипломного проектирования на выпускающей кафедре. Предложенное решение упрощает процесс согласования тем дипломных проектов, выбора руководителей дипломного проекта, подготовки документации по преддипломной практике и дипломному проектированию для ответственного за данные процессы, а также позволяет осуществлять контроль за ходом дипломного проектирования посредством электронного журнала прохождения контрольных точек и ключевых процессов дипломного проектирования вплоть до защиты дипломных проектов. Веб-приложение разработано на кафедре информационных систем и технологий с учетом многолетнего опыта организации и проведения данного процесса. Разработанное программное средство позволит повысить эффективность работы руководителей дипломных проектов и ответственного за преддипломную практику и дипломное проектирование, увеличит оперативность, достоверность и наглядность информации, получаемой в ходе дипломного проектирования.

Ключевые слова. Дипломное проектирование, бизнес-процесс, электронный журнал, контроль хода процесса, цифровизация образования, автоматизация процесса, клиент-серверное приложение.

Введение. Целью данного проекта является разработка программного средства, предназначенного для автоматизации и цифровизации процесса организации и сопровождения преддипломной практики и дипломного проектирования, и, как следствие, повышения эффективности работы руководителей дипломных проектов и ответственного за преддипломную практику и дипломное проектирование, увеличения оперативности, достоверности и наглядности информационных потоков, повышения удобства предоставления и получения информации студентам и, в целом, упрощения организационных процессов.

Основная часть. Для решения поставленной цели разрабатываемое веб-приложение обладает тремя основными уровнями доступа: «студент», «преподаватель» (руководитель дипломного проекта, консультант от кафедры, нормоконтролер, консультант по технико-экономическому обоснованию) и «ответственный за дипломное проектирование». Каждый уровень доступа имеет свой набор доступных функций. Ответственный за дипломное проектирование по своей сути является администратором приложения и всех бизнес-процессов.

Основные функции для уровня доступа «студент»:

- заполнение информации по преддипломной практике;
- заполнение информации по дипломному проекту;
- выбор руководителя дипломного проекта из штатных сотрудников кафедры или отправка запроса на утверждение внешнего руководителя;
- просмотр информации о дипломном проекте и прохождении контрольных точек;
- отправка сообщений на электронную почту руководителя;
- формирование заявления на утверждение темы дипломного проекта и руководителя;
- просмотр объявлений от ответственного за дипломное проектирование.

Основные функции для уровня доступа «преподаватель»:

- просмотр информации о дипломных проектах студентов-дипломников;
- отправка сообщений на почту студенту;
- редактирование информации о дипломном проекте (темы, задания);
- проставление результатов прохождения контрольных точек (электронный журнал хода дипломного проектирования);
- просмотр объявлений от ответственного за дипломное проектирование.

Основные функции для уровня доступа «ответственный за дипломное проектирование»:

- просмотр информации обо всех дипломных проектах;
- изменение информации о дипломных проектах;
- рассмотрение поступающих запросов на утверждение внешних руководителей;
- конфигурация информации о сроках прохождения контрольных точек, нормоконтроля, рабочих комиссий, сдачи дипломных проектов;
- импорт данных (база студентов выпускного курса, штатных сотрудников кафедры и др.) из электронных таблиц;
- экспорт данных в электронные документы;

- формирование отчетов о начале/итогах преддипломной практики;
- формирование проекта приказа об утверждении тем и руководителей дипломных проектов;
- назначение нормоконтролеров, консультантов от кафедры, консультантов по технико-экономическому обоснованию, председателей и членов рабочих комиссий, дат защиты дипломных проектов;
- отправка уведомлений преподавателям и студентам.

Диаграмма вариантов использования разработанного веб-приложения приведена на рисунке 1.

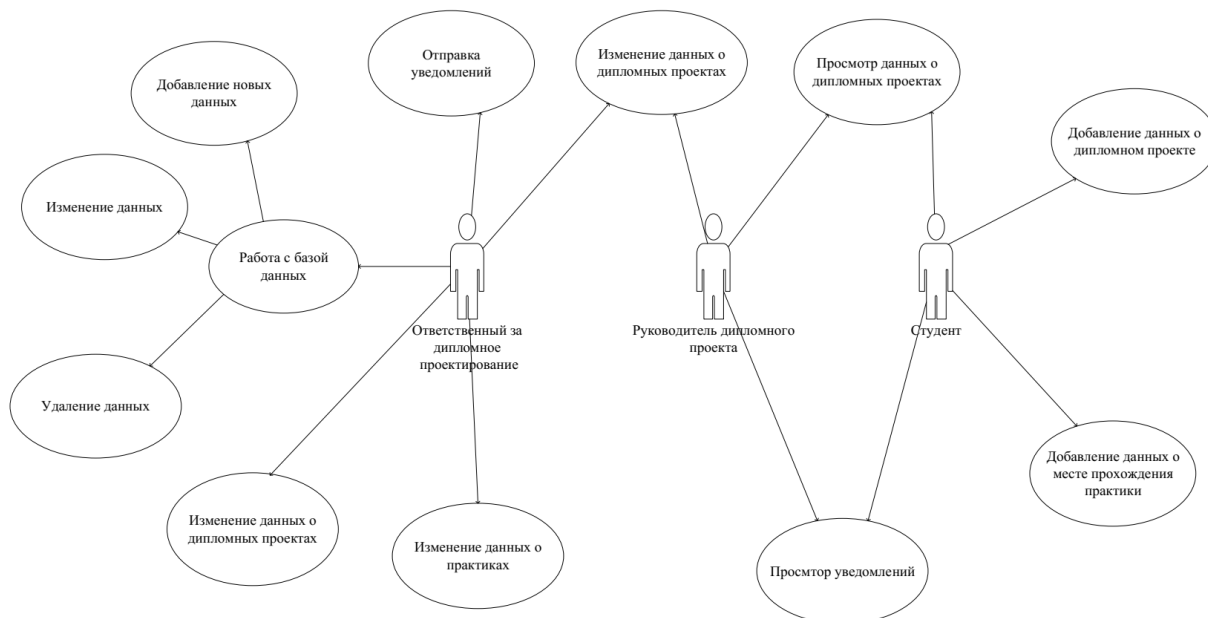


Рисунок 1 – Диаграмма вариантов использования приложения

Доступ к реляционной базе данных осуществляется по глобальной сети Интернет, непосредственно через страницы веб-платформы, что позволяет осуществлять прямой доступ к информации одновременно множеству пользователей, находящимся на любом расстоянии от сервера, хранящего данные, а также позволяет вносить изменения в базу данных максимально простым и эффективным способом. Для защиты информации, сохранения её конфиденциальности и разграничения прав доступа используется механизм авторизации и аутентификации.

Для разработанного программного средства была выбрана клиент-серверная архитектура.

Серверная часть приложения спроектирована на основе шаблона Model-View-Controller (MVC). MVC – шаблон программирования, разделяющий архитектуру приложения на три модуля: модель (Model), представление (View), контроллер (Controller). Он позволяет изменять каждый компонент независимо друг от друга для простой разработки и поддержки веб-приложений [1]. Серверная часть представляет собой Web Api ASP.NET приложение, написанное на языке программирования C#.

Пользовательская часть реализована на базе JavaScript-библиотеки React. Веб-приложения, написанные с помощью библиотеки React, позволяют менять контент страницы без последующей перезагрузки. С помощью дополнительных библиотек можно полностью управлять пользовательской частью приложения. Для маршрутизации в веб-приложении используется библиотека «react-router-dom». Для хранения информации приложения используется библиотека «react-redux». Для реализации компонент приложения используется «material ui» (MUI). Отправка запросов на сервер реализована с помощью JavaScript функции «fetch».

За связь между пользовательской и серверной частями приложения отвечают контроллеры, обрабатывающие входящие HTTP-запросы по определенному маршруту в MVC-приложениях.

В общем виде алгоритм обработки запросов представлен в следующем виде:

- 1) приложение получает запрос;
- 2) система маршрутизации выбирает необходимый контроллер, который обработает запрос;
- 3) контроллер обрабатывает запрос, обращаясь к моделям данным;
- 4) контроллер выполняет роль связующего звена между представлениями и моделями;
- 5) клиент получает результат обработки.

Важнейшей функциональной задачей разработанного приложения является подготовка и формирование отчетной документации бизнес-процессов для ответственного за дипломное проектирование и преддипломную практику от кафедры. По соответствующему запросу необходимая информация извлекается из базы данных и формируется документы установленного образца в формате .docx такие как: отчет о начале преддипломной практики, отчет об окончании преддипломной практики, проект приказа об утверждении тем и руководителей дипломных проектов, проект распоряжения о составе и графике работы рабочих комиссий, проект распоряжения о

назначении рецензентов дипломных проектов. Формирование отчетной документации происходит в один клик по выбранной форме обучения. Экранная форма приложения с примером формирования отчетов представлена на рисунке 2.



Рисунок 2 – Экранная форма страницы формирования отчетной документации приложения

Назначение пользователей роли «преподаватель» для выполнения функций нормоконтролера, консультанта по технико-экономическому обоснованию для групп студентов выполняет ответственный за дипломное проектирование. Также он формирует состав и график работы рабочих комиссий, закрепляет за каждым дипломных проектом рецензента посредством инструментов своего личного кабинета. Также ответственному за дипломное проектирование доступна информация по каждому дипломному проекту всех студентов и электронный журнал прохождения контрольных точек дипломного проектирования (опроцентовок).

Руководителям дипломных проектов доступен для заполнения электронный журнал только по их студентам-дипломникам. Пример электронного журнала приведен на рисунке 4.

Диаграмма деятельности разработанного программного средства представлена на рисунке 5.

Разработанное приложение удовлетворяет необходимым параметрам безопасности данных и защиты персональных данных пользователей. В приложении невозможно разместить свои персональные данные для обработки, не дав на это согласия. Кроме того, пользователями уровня «студент» и «преподаватель», относящийся к категории штатного сотрудника, могут быть зарегистрированы только те лица, которые имеются в базе данных. База данных студентов и штатных сотрудников кафедры наполняется ответственных за дипломное проектирование. Для этого предусмотрен ручной ввод, а также импорт из электронных таблиц, которые можно получить из сторонних баз данных студентов и сотрудников. Заявки на утверждение внешних дипломных руководителей рассматриваются и принимаются либо отклоняются в личном кабинете ответственного за дипломное проектирование. В любом из случаев на электронную почту студентов и руководителей отправляются соответствующие уведомления.

The screenshot shows the 'Дипломные проекты' section of the application. It displays a table for a project titled 'Хотак Сабур Абдурахманович / группа: 981063 / ВФО'. The table has columns for 'Место прохождения практики', 'Город', 'Тема дипломного проекта', 'Руководитель', 'Должность, степень и звание', 'Почта и номер телефона', 'Консультант от кафедры', 'Должность, степень и звание', and 'Почта и номер телефона'. Below this, there is a progress table with columns for 'Первая опроцентовка 30% (Защита ПДП)', 'Вторая опроцентовка 60%', 'Третья опроцентовка 80%', 'Четвертая опроцентовка', 'Нормо-контроль', 'ТЭО', 'ПК', and 'Защита'. The progress table shows dates and percentages for each stage, with a total score of 8 and a '+' sign. There is an 'ИЗМЕНИТЬ' button below the table.

Место прохождения практики	Город	Тема дипломного проекта	Руководитель	Должность, степень и звание	Почта и номер телефона	Консультант от кафедры	Должность, степень и звание	Почта и номер телефона
ООО ДЕВСТРИМ	Минск	ВЕБ-ПРИЛОЖЕНИЕ "ВЫПУСКАЮЩАЯ КАФЕДРА" НА ПЛАТФОРМЕ .NET С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ REACT.JS	Савенко Андрей Геннадьевич	Старший преподаватель кафедры ИСИТ, м.т.н.	savenko@bsuir.by +375291111111	Савенко Андрей Геннадьевич	Старший преподаватель кафедры ИСИТ, м.т.н.	savenko@bsuir.by +375291111111
Первая опроцентовка 30% (Защита ПДП)	Вторая опроцентовка 60%	Третья опроцентовка 80%	Четвертая опроцентовка	Нормо-контроль	ТЭО	ПК	Защита	
с 26.12.2022 по 26.12.2022	с 26.12.2022 по 26.12.2022	с 26.12.2022 по 26.12.2022	с 26.12.2022 по 26.12.2022	с 26.12.2022 по 26.12.2022	с 26.12.2022 по 26.12.2022	с 26.12.2022 по 26.12.2022	26.12.2022	26.12.2022
Оценка: 8	60%	75%	100%	+	-	+		

Рисунок 4 – Электронный журнал прохождения контрольных точек

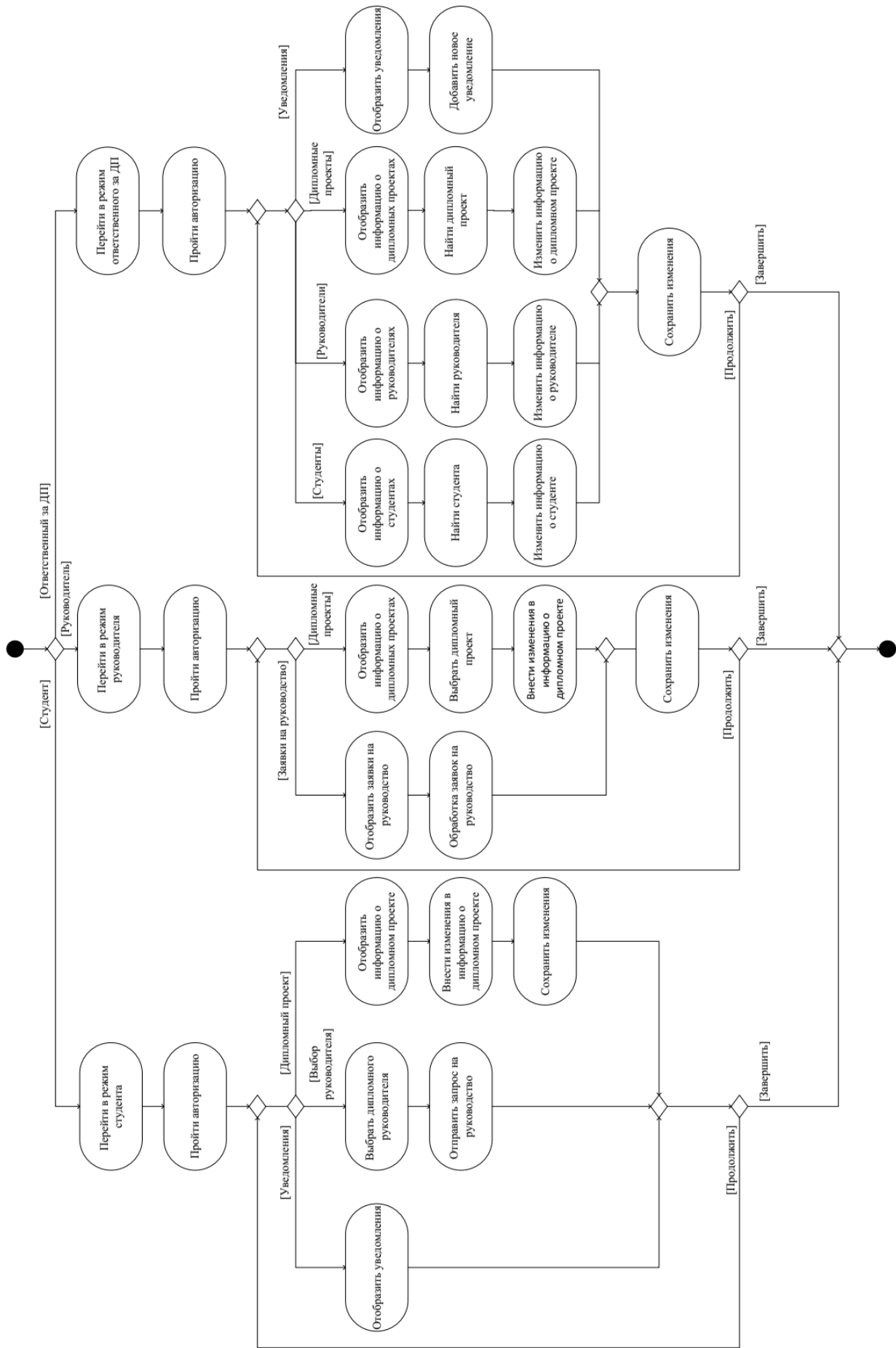


Рисунок 5 – Диаграмма деятельности программного средства

Заключение. Таким образом, было разработано и внедрено на выпускающей кафедре информационных систем и технологий ИИТ БГУИР веб-приложение для автоматизации и цифровизации процесса дипломного проектирования. Данное программное средство значительно упрощает и делает удобным процесс организации, сопровождения и прохождения дипломного проектирования для всех участников данного бизнес-процесса, минимизирует ошибки, обусловленные человеческим фактором, что особо важно для кафедры, выпускающей большое количество студентов.

Разработанное приложение разрешит пользователю-регистратору записать пациента на прием, заполнить данные о пациенте и об оказанных услугах, указать их стоимость и данные персонала. А у администратора есть доступ для изменения данных во всех таблицах: «Пациенты», «Сотрудники» и «Услуги». Основные уникальные функции, отличающие средство от аналогов, заключаются в доступности и простоте пользования приложения, задействовании небольшого объема памяти компьютера, современном дизайне, высокой скорости работы программы, отсутствии обязательной регистрации для работы с приложением.

Список использованных источников:

1. SkillFactory [Электронный ресурс]. – Режим доступа : <https://blog.skillfactory.ru/glossary/mvc>

UDC 004.42

WEB-APPLICATION "THE GRADUATE DEPARTMENT" FOR AUTOMATION AND DIGITALIZATION OF GRADUATION DESIGN

Khotak S.A, Savenko A.G.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Savenko A.G. – Master of Engineering Sciences

Annotation. This paper presents a developed web application designed to automate and digitalize the process of graduation design at the graduating department. The proposed solution simplifies the process of coordinating the themes of graduation projects, selecting the leaders of the graduation project, preparing documentation on undergraduate practice and graduation design for the person responsible for these processes, and also allows you to monitor the progress of graduation design through an electronic log of passing checkpoints and key processes of graduation design up to defense of diploma projects. The web application was developed at the Department of Information Systems and Technologies, taking into account many years of experience in organizing and conducting this process. The developed software tool will increase the efficiency of the work of the leaders of graduation projects and those responsible for undergraduate practice and graduation design, will increase the efficiency, reliability and visibility of information obtained during graduation design.

Keywords. Graduation project, business process, e-journal, process control, digitalization of education, process automation.

УДК 004.75+004.4

ТРАНСФОРМАЦИЯ ПРИЛОЖЕНИЙ ОТ МОНОЛИТНОЙ К МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ

Ширшов А. А., студент

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Парамонов А.И. – канд. техн. наук, доцент, зав. каф. ИСиТ

Аннотация. Современные программные средства требуют быстрой развёртки в облачных хранилищах, легкого и удобного масштабирования, что невозможно при использовании монолитной архитектуры. Обозначены основные проблемы монолитной архитектуры и способы их возможных решений. Предлагаются пути перехода к микросервисной архитектуре.

Ключевые слова. Микросервис, монолит, архитектура, программное обеспечение, облачные вычисления, API, масштабирование.

Введение. Потребность в микросервисной архитектуре приложений возникла совсем не спонтанно. Цифровая трансформация сегодня активно входит во все отрасли экономики, что влечет изменение не только технологических процессов бизнеса, но и самих инструментов трансформации. ИТ-инфраструктура становится с каждым годом всё сложнее, обрастает новыми системами и сервисами. Объемные приложения с большим количеством взаимодействующих между собой функциональных модулей, большие команды разработчиков, создающих проект, требования к скорейшему выпуску и своевременному обновлению программного обеспечения (ПО), необходимость периодического масштабирования отдельных модулей – всё это предпосылки к тому, чтобы серьезно задуматься о разработке приложений на микросервисной архитектуре с нуля или перестраивании на микросервисы уже существующей монолитной структуры [1].

В то же время, отношение бизнеса и даже ИТ-сообщества к микросервисной архитектуре разработки приложений сегодня не столь однозначно и возможно даже противоречиво, как в свое время к RPA [2]. Энтузиасты называют монолитные приложения атавизмами уходящей эпохи и спешно переходят на микросервисы. Однако, следует отметить, что не всем, кто принял решение о переходе на микросервисы, удается окупить вложения в них.

Основная часть. Монолитная архитектура – это традиционная модель ПО, которая представляет собой единый модуль, работающий автономно и независимо от других приложений. Монолитом часто называют нечто большое и неповоротливое, и эти два слова хорошо описывают монолитную архитектуру для проектирования программного продукта. Монолитная архитектура – это отдельная большая вычислительная сеть с единой базой кода, в которой объединены все бизнес-задачи. Упрощенная схема такой архитектуры приведена на рисунке 1.

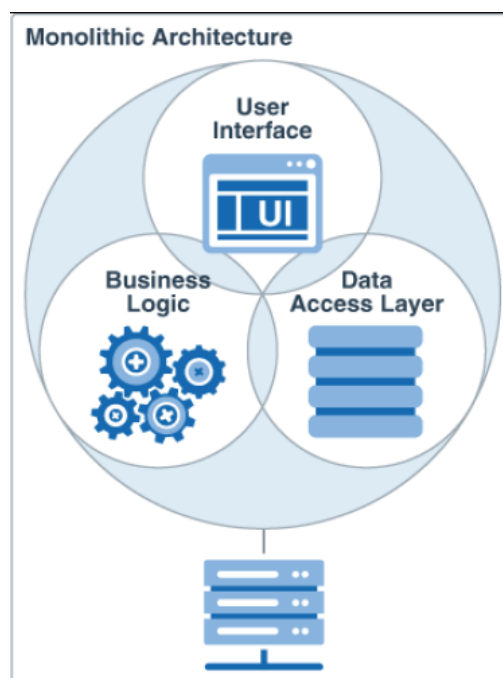


Рисунок 1 – Упрощенная схема монолитной архитектуры

«Монолиты» удобно использовать на начальных этапах проектов, чтобы облегчить развертывание и не тратить слишком много умственных усилий при управлении кодом. Это позволяет сразу выпускать все, что есть в монолитном приложении. Однако, чтобы внести изменения в такое приложение, необходимо обновить весь стек через базу кода, а также создать и развернуть обновленную версию интерфейса, находящегося на стороне службы. Это ограничивает работу с обновлениями и требует много времени. Проблема при разработке и поддержке ПО с монолитной архитектурой может возникнуть, в частности, в момент слияния веток проекта (merge conflict), когда несколько программистов одновременно модифицируют один и тот же участок кода. Этот недостаток не критичен, но увеличивает время разработки, особенно если в приложение добавляется новый модуль: например, функциональность доставки товара со склада интернет-магазина дополняется модулем закупки или аналитики.

Проблема масштабируемости приложения появляется, когда количество его пользователей начинает расти, причем выстреливший стартап зачастую демонстрирует очень активную динамику. Некоторые модули монолита могут иметь специальную архитектуру, которая будет обеспечивать высокие нагрузки, но масштабировать в любом случае придется всё приложение целиком. В монолите мы не можем отдельно поменять архитектуру только высоконагруженных модулей [3]. Эту ситуацию можно образно сравнить, к примеру, с развивающейся сетью магазинов, у которых единая служба бухгалтерии: в случае монолита при масштабировании приложения нам пришлось бы в каждый из этих магазинов добавить своего бухгалтера, при этом с ростом количества покупателей нагрузка на этот отдел осталась бы прежней.

Микросервисная архитектура — распространенный подход к разработке ПО, когда приложение разбивается на небольшие автономные компоненты («микросервисы») с четко определенными интерфейсами. Именно эта архитектура характерна для Cloud-native приложений, которые сейчас популярны благодаря преимуществам облачных сред и их возможностям для бизнеса. В микросервисной архитектуре слабо связанные сервисы взаимодействуют друг с другом для выполнения задач, относящихся к их бизнес-возможностям. Микросервисы в значительной степени получили свое название из-за того, что сервисы здесь меньше, чем в монолитной среде. Но нужно понимать, что приставка «микро» — об указании на бизнес-возможности (функционал), а не на размер. Упрощенная схема такой архитектуры представлена на рисунке 2.

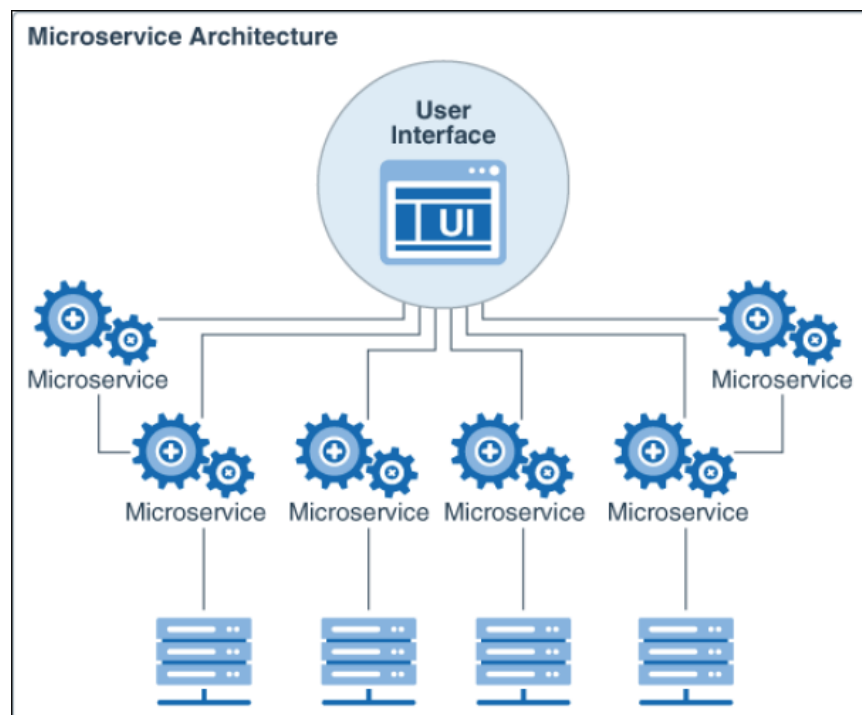


Рисунок 2 – Упрощенная схема микросервисной архитектуры

Желание заменить монолитную архитектуру на микросервисы основано на статистике, которая показывает, что между монолитными и микросервисными системами существует значительная разница в производительности, которая порой достигает 79,2% [4]. Статистика [5] показывает, что:

- 68% компаний уже используют микросервисы в производстве и разработке;
- 36% крупных компаний, 50% средних компаний, 44% малых компаний используют микросервисы в производстве и разработке;
- 26% компаний рассматривают микросервисы, но еще не начали их внедрять;
- команды, которые перешли на микросервисы, отчитались о 13-кратном увеличении частоты выпусков ПО.

Стоит отметить, что при переходе с «монолита» на «микросервисы» интеграция сталкивается с некоторыми проблемами. Обычно это означает, что разработчикам нужно восстанавливать прежние версии с нуля. По этой причине весь процесс может также привести к определенным сложностям, которые связаны с тратой ресурсов (времени и денег), отсутствием требований или фактического состояния (только код), возникновением технологических вопросов (по технологиям, подходам, внедрению и т.п.). Ключевые моменты в сравнении архитектур представлены в таблице 1.

Таблица 1 – Сравнение монолитной и микросервисной архитектуры.

Монолит	Микросервисы
Типичное приложение состоит из базы данных, клиентского пользовательского интерфейса и серверного приложения.	В основном получили свое название из-за того, что сервисы здесь меньше, чем в монолитной среде.
Все части ПО объединены, так что управлять всеми его функциями можно в одном месте.	Сервисы связаны слабо, взаимодействуют между собой и автономно выполняют бизнес-задачи.
Компоненты ПО взаимосвязаны и взаимозависимы.	Существует много единиц развертывания. Каждый сервис разрабатывается и разворачивается независимо друг от друга.
Приложения обеспечивают быструю связь между программными компонентами благодаря общему коду и памяти.	Меньшие размеры сервисов помогают, когда дело доходит до времени компиляции, выполнения и времени запуска тестов.

Заключение. Трансформация архитектуры приложений и переход от монолита к микросервису предполагает два основных пути решения:

1. Переписать приложение с нуля на микросервисную архитектуру и использовать лучшие практики написания кода. На это решение потребовалось бы более года разработки, а также команда из пяти и более человек.

2. Внедрить один микросервис, который покрыл бы небольшую, но наиболее часто используемую часть функциональности, из-за которой возникают самые большие (или частые) проблемы с производительностью. На это потребовалось бы около двух месяцев разработки и команда всего из двух человек.

Для успешного начала необходимо внедрить логику в микросервис, создав API Gateway, благодаря которому можно было легко общаться обеим сторонам — клиентам существующих проектов и бизнес-клиентам. Далее постепенно переносить бизнес-логику в отдельные микросервисы и подключать их к Gateway, чтобы агрегировать вызовы микросервисов.

Список использованных источников:

1. Переход от монолита к микросервисам: когда опыт разработчиков трансформируется в бизнес-результат. [Электронный ресурс] Мир цифровых и информационных технологий. Режим доступа: <https://www.it-world.ru/tech/practice/173784.html>. (дата обращения: 03.04.2023).
2. Бадмаева А.Д., Перерва О.Л. Риски внедрения технологии RPA на наукоемкое предприятие // Научный результат. Экономические исследования. 2020. №3. [Электронный ресурс] Режим доступа: <https://cyberleninka.ru/article/n/riski-vnedreniya-tehnologii-gra-na-naukoeemkoe-predpriyatie> (дата обращения: 03.04.2023).
3. Переход от монолита к микросервисам: история и практика. [Электронный ресурс] Блог компании Райффайзен Банк. Режим доступа: <https://habr.com/ru/company/raiffeisenbank/blog/458404/>. (дата обращения: 03.04.2023).
4. От устаревшего монолита - к микросервисам: как решиться на миграцию. [Электронный ресурс] Режим доступа: <https://highload.today/blogs/ot-ustarevshego-monolita-k-mikroservisam-kak-reshitsya-na-migratsiyu/>. (дата обращения: 03.04.2023).
5. Отчет O'Reilly о внедрении микросервисной архитектуры в 2020 году. [Электронный ресурс] Режим доступа: <https://www.businesswire.com/news/home/20200716005101/en/O%E2%80%99Reilly%E2%80%99s-Microservices-Adoption-in-2020-Report-Finds-that-92-of-Organizations-are-Experiencing-Success-with-Microservices>. (дата обращения: 03.04.2023).

UDC 004.75+004.4

APPLICATION TRANSFORMATION FROM MONOLITHIC TO MICROSERVICES ARCHITECTURE

Shirshov A. A.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Paramonov A.I. – Candidate of Engineering Sciences, Associate Professor

Annotation. Modern software tools require fast deployment in cloud storages, easy and convenient scaling, which is impossible when using a monolithic architecture. The main problems of monolithic architecture and ways of their possible solutions are outlined. Ways of transition to microservice architecture are offered.

Keywords. Microservice, monolithic architecture, software, cloud computing, API, scaling.

УДК 621.3.07+551.583

СИСТЕМА УПРАВЛЕНИЯ МИКРОКЛИМАТОМ ПРИ АККЛИМАТИЗАЦИИ РАСТЕНИЙ

Шутов Д.О., студент

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Шпак И.И. – канд. техн. наук, доцент каф. ИСиТ

Аннотация. В статье приведены результаты, полученные в процессе создания автоматизированной системы управления микроклиматом при акклиматизации растений. Разработанная система позволяет поддерживать заданные параметры микроклимата, что помогает растениям легче акклиматизироваться в новых условиях. Автором статьи выполнено схемотехническое проектирование системы, разработаны алгоритм и программное обеспечение ее функционирования, а также конструкция печатного узла, с применением современной САПР.

Ключевые слова. Акклиматизация, микроклимат, контроль температуры и влажности, микроконтроллер, схемотехническое проектирование, разработка программного обеспечения.

Введение. Акклиматизация растений - приспособление растений к климату, отличному от климата их родины. Понятие климата включает в себя совокупность таких параметров как температура, влажность, количество осадков, наличие и продолжительность ветров, длительность светового дня и ряд других факторов. Жизнь растений определяется не только климатическими, но и почвенными факторами, а также качествами почвенных вод. Влияния этих факторов могут не совпадать друг с другом; но иногда из различных условий климата и свойств почвы может получиться новая комбинация, делающая возможным не только существование, но и плодоношение растений. Процесс акклиматизации затрагивает растение целиком. Осваиваться в новых условиях приходится всем его частям. Крона адаптируется к изменению спектра света и сокращению освещенности, уменьшению влажности и температуры. Корням же приходится привыкать к новому составу воды, почвы и подкормок. Для этого растение мобилизует все свои силы и на некоторое время его рост прекращается или замедляется. Когда же привыкание успешно произошло, рост возобновится и, даже, существенно ускорится. [1]

Актуальность разрабатываемого устройства заключается в том, что растения могут полноценно произрастать только в определенных климатических условиях и при переносе в иные климатические условия не всегда способны самостоятельно в них акклиматизироваться. Использование искусственных экосистем с системой поддержания микроклимата позволяет создать для этих растений условия, приближенные к необходимым им и, таким образом, позволяет постепенно акклиматизировать культуры в условиях, неподходящих для выращивания этих культур.

Система управления микроклиматом предназначена для автоматизации процесса поддержания в экосистеме необходимых климатических параметров. Оно представляет из себя контроллер, к которому подключаются датчики, измеряющие эти параметры, такие как датчик температуры и влажности воздуха, влажности почвы. Система управления, постоянно находясь в рабочем режиме, отслеживает показания датчиков. В зависимости от этих показаний устройство приводит в действие те или иные исполнительные устройства (обогреватель, увлажнитель, привод открытия дверей) для того, чтобы привести параметры к необходимым значениям. [2]

Система может быть установлена непосредственно внутри помещения либо в комнате, откуда осуществляется управление экосистемой, а в помещении, где находится экосистема, могут располагаться только датчики и исполнительные устройства.

Основная часть. Разработанное устройство предназначено для считывания и обработки выходных параметров датчиков, расположенных в теплице, выработки управляющих сигналов на исполнительные устройства и последующую индикацию параметров. Структурная схема системы управления микроклиматом при акклиматизации растений представлена на рисунке 1.

Для измерения таких физических величин как: температура воздуха, влажность воздуха и влажность почвы, с помощью датчиков снимаются нужные показатели и полученные данные отправляются в блок управления. Блок управления снимает и записывает эти данные; далее он обрабатывает их и выдает сигнал на исполнительные устройства, а также на устройство отображения (дисплей). Блок питания обеспечивает устройство стабилизированным питанием.

Для измерения таких физических величин как: частота вращения, пусковой момент, сила фазных токов и напряжение, температура испытуемого электродвигателя, с помощью датчиков снимаются нужные показатели и полученные данные отправляются в блок управления. Блок управления снимает и записывает эти данные; далее он сопоставляет полученные данные со значениями, заложенными в программе, и выдает информацию на устройство отображения

(дисплей). Блок питания обеспечивает устройство стабилизированным питанием. Устройство торможения обеспечивает требуемую нагрузку на двигатель. Посредством блока коммутации блок управления запускает или останавливает испытуемый двигатель.

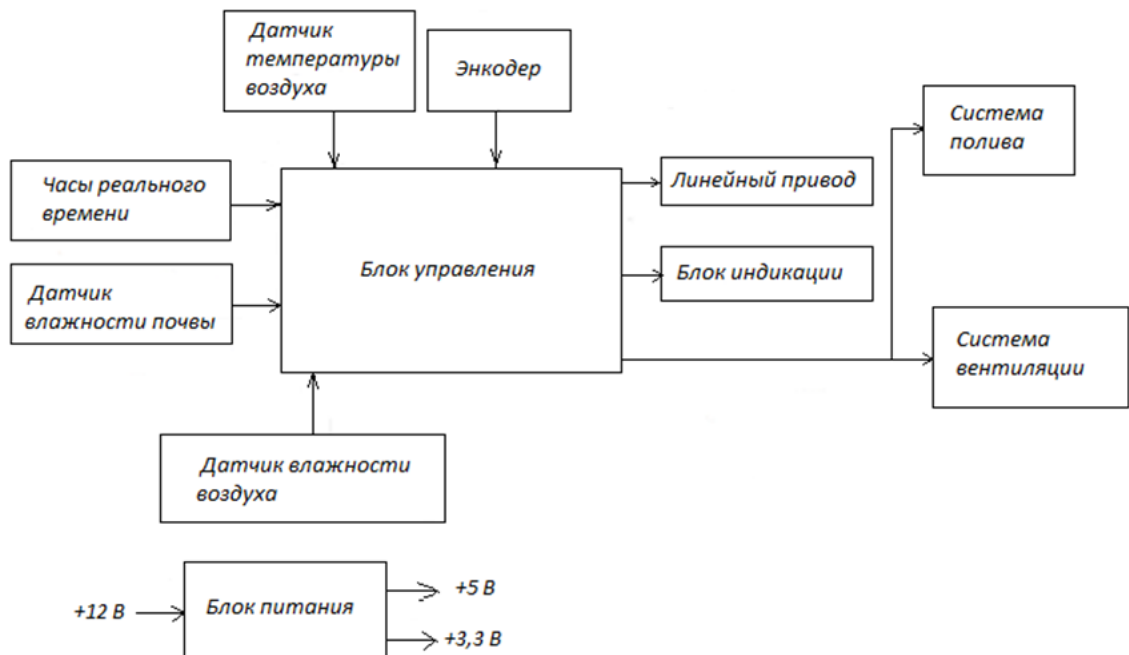


Рисунок 1 – Схема электрическая структурная устройства

Ядром устройства, является блок управления на основе микроконтроллера, который отвечает за получение и обработку поступающих сигналов. Основой блока управления является микроконтроллер. Был выбран микроконтроллер серии ATmega [3], архитектура которого представлена на рисунке 2.

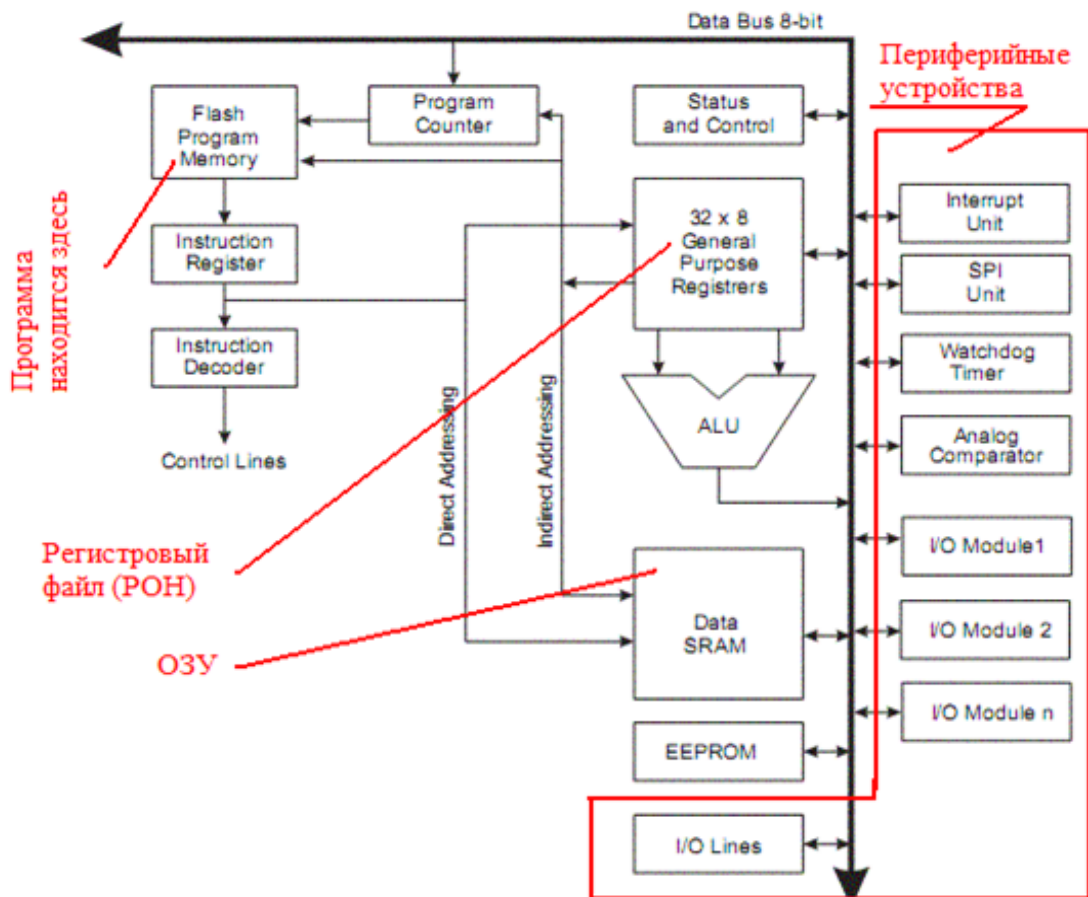


Рисунок 2 – Архитектура микроконтроллера ATmega328

Выбранный тип контроллера пользуется большой популярностью в мире благодаря невысокой цене, стабильности параметров и большому количеству информационных ресурсов [4], которые позволяют освоить его в кратчайшие сроки и с небольшими финансовыми издержками.

Для связи с периферийными устройствами в микроконтроллере используются порты ввода-вывода и необходимые интерфейсы. Для подключения датчика температуры используется интерфейс 1Wire. Для подключения датчиков с аналоговыми сигналами используем встроенный АЦП и порты ввода-вывода. Клавиатура и дисплей также подключены к портам ввода-вывода.

Постоянное напряжение поступает на блок питания и преобразуется в постоянные питающие напряжения 5 и 3,3 В.

Весьма непросто в процессе проектирования оказался выбор датчика температуры и влажности воздуха [5,6,7]. Существует множество таких датчиков, однако наиболее широко применяются три: DHT22, HTU21 и BME280.

Самым предпочтительным оказался цифровой датчик температуры, влажности и давления производства Bosch – BME280. Этот прецизионный датчик может измерять относительную влажность воздуха от 0 до 100% с точностью $\pm 3\%$, атмосферное давление от 300 гПа до 1100 гПа с абсолютной точностью ± 1 гПа, и температуру от -40°C до 85°C с точностью $\pm 1,0^{\circ}\text{C}$. Модуль поставляется со встроенным стабилизатором напряжения 3,3 В LM6206 и преобразователем уровней напряжения на шине I2C, поэтому его можно без проблем использовать с микроконтроллером, работающим с логическими уровнями 3,3 В или 5 В.

Переменное сетевое напряжение поступает на блок питания и преобразуется в постоянные питающие напряжения.

Блок коммутации осуществляет включения-выключения испытуемого двигателя посредством реле. Для защиты микроконтроллера от всплесков индуктивного напряжения блок коммутации выполнен с использованием оптопары, для осуществления гальванической развязки. Реле непосредственно управляет состоянием испытуемого электродвигателя.

Весьма непросто в процессе проектирования оказался выбор датчика тока. Существует множество методов измерения тока [5,6], однако в промышленности наиболее широко применяются три: на основе использования токового шунта, на основе эффекта Холла и трансформатора тока.

Самой простой является схема измерения тока с помощью шунта. При этом измеряется падение напряжения на шунте, после чего рассчитывается значение тока. Основными недостатками здесь являются:

- отсутствие гальванической развязки;
- низкий КПД.

Указанные недостатки существенно ограничивают использование токовых шунтов.

Схемы для измерения тока на основе трансформатора тока, с помощью датчика на эффекте Холла и измерения напряжения с помощью делителя напряжения показаны на рисунке 3.

Схема подключения выбранного датчика температуры и влажности BME280 показана на рисунке 3.

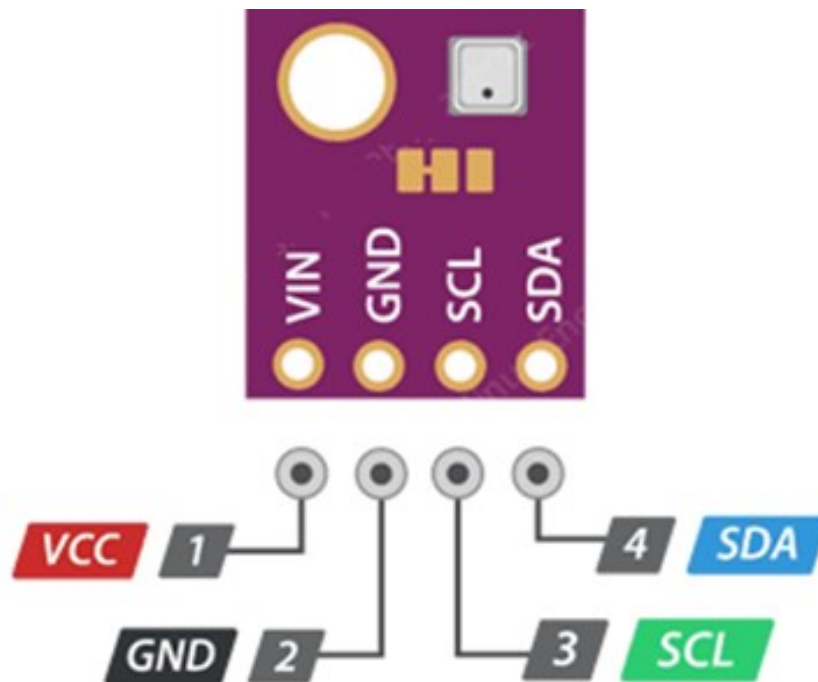


Рисунок 3 – Схема выводов датчика температуры и влажности BME280

Заключение. Для реализации разработанного устройства была выбрана современная элементная база, и проведён расчёт функциональных узлов. Поэтапно были разработаны алгоритм работы блока управления и программное обеспечение [4] для микроконтроллера.

Отличительной особенностью разработки является эффективное использование результатов научных исследований, выполненных по данной проблеме [8].

Конструкция блока управления была реализована в виде небольшого блока. В процессе проектирования топологии печатной платы и печатного узла на её основе была использована система автоматизированного проектирования Altium Designer.

Одним из преимуществ созданной системы управления микроклиматом является сравнительная простота его настройки, невысокая стоимость, а также возможность улучшения эксплуатационных характеристик и создания более сложных систем контроля и управления микроклиматом.

Список использованных источников:

5. 1. Маркетинговая стратегия частной стоматологической клиники [Электронный ресурс]. – Режим доступа: <https://works.doklad.ru/view/58jfb1ChrMM.html>.

6. ProfmanDental: программа по управлению частной стоматологией [Электронный ресурс]. – Режим доступа: <https://profmandental.by/>.

7. StomX: программа для стоматологии [Электронный ресурс]. – Режим доступа: <https://stomx.ru/>

8. Dental4Windows: программа для стоматологии [Электронный ресурс]. – Режим доступа: <https://d4w.ru/>.

UDC 621.3.07+551.583

MICROCLIMATE MANAGEMENT SYSTEM FOR ACCLIMATIZATION OF PLANTS

Shutov D.O.

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Shpak I.I. – Candidate of Engineering Sciences, Associate Professor

Annotation. The article presents the results obtained in the process of creating an automated microclimate management system for acclimatization of plants. The developed system allows you to maintain the specified parameters of the microclimate, which helps plants to acclimatize more easily in new conditions. The author of the article performed circuit design of the system, developed an algorithm and software for its functioning, as well as the design of the printing unit, using modern CAD..

Keywords. Acclimatization, microclimate, temperature and humidity control, microcontroller, circuit design, software development.

АВТОРСКИЙ УКАЗАТЕЛЬ

Бич М.В.	5
Варченя И.В.	8
Величко В.М.	13
Гриневич А.И.	16
Дунай Р.Н.	20
Желенок Д.А.	25
Колдунова А.А.	31
Курилец А.С.	35
Ладычук И.С.	31
Савенко А.Г.	8, 25, 68
Самсонов Н.Н.	39
Сапега Я.Е.	42
Северин К. М.	45
Скудняков Ю.А.	13, 53, 56, 60, 64
Славинский Г.А.	48
Тыманович Н.А.	53, 56, 60, 64
Хотак С.А.	68
Ширшов А. А.	73
Шутов Д.О.	76

Научное издание

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

**Сборник статей 59-й научной конференции
аспирантов, магистрантов и студентов
(Минск, 17-21 апреля 2023 года)**

В авторской редакции

Ответственный за выпуск: А.И. Парамонов

Компьютерная верстка: Д.М. Карнаух