

УДК 612.087.1

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДЛЯ ПРОВЕДЕНИЯ ИССЛЕДОВАНИЙ МЕДИЦИНСКИХ ИМПЛАНТИРУЕМЫХ ПРИБОРОВ



А.Н. Осипов
Заведующий лабораторией
НИЧ БГУИР, кандидат
технических наук, доцент
osipov@bsuir.by



С.П. Урбанович
Ведущий инженер
Отраслевой лаборатории
ЦМИ НИЧ БГУИР
urbanovich@bsuir.by



М.П. Батура
Заведующий лабораторией
НИЧ БГУИР, доктор
технических наук, профессор
bmpbel@bsuir.by



А.А. Ивановский
Студент БГУИР
a.ivanovskiy03@mail.ru



И.О. Хазановский
Аспирант БГУИР
khaz_igor@bsuir.by



А.В. Короткевич
Доцент кафедры
микроэлектроники БГУИР,
кандидат технических наук,
доцент
korotkevich@bsuir.by

А.Н. Осипов

Окончил Минский радиотехнический институт. Область научных интересов – биомедицинская инженерия.

С.П. Урбанович

Окончил Белорусский государственный университет информатики и радиоэлектроники. Специалист в области синтеза и цифровой обработки сигналов

М.П. Батура

Окончил Минский радиотехнический институт. Область научных интересов - системный анализ, управление и обработка информации в технических системах

А.А. Ивановский

Студент Белорусского государственного университета информатики и радиоэлектроники. Область научных интересов связана с разработкой методов и алгоритмов построения информационно-компьютерных систем.

И.О. Хазановский

Окончил Военную Академию Республики Беларусь. Область научных интересов связана с биомедицинской инженерией.

А.В. Короткевич

Окончил Минский радиотехнический институт. Область научных интересов - технология формирования изделий микро-, нано- и криоэлектроники с использованием анодных оксидов вентильных металлов.

Аннотация. Рассмотрены подходы к созданию автоматизированной системы для исследований медицинских имплантируемых приборов. Описаны архитектура программного обеспечения, включающего взаимодействие с аппаратной платформой по протоколу *Modbus*, управление датчиками и исполнительными устройствами, а также способ хранения и обработки данных с использованием СУБД (Система управления базами данных). Показано, что модульная структура программно-аппаратного комплекса и использование современных инструментов позволяют адаптировать систему под различные сценарии испытаний, обеспечить хранение и обработку большого количества данных исследований.

Ключевые слова: автоматизированная система, медицинские имплантируемые приборы, ускоренные испытания, протокол *Modbus*, *Raspberry Pi CM4*, *PyQt5*, управление параметрами эксперимента, хранение и обработка данных.

Введение. Устройства, имплантируемые в тело человека, играют важную роль как в поддержании жизнедеятельности, так и в диагностике различных физиологических параметров, таких как уровень глюкозы, давление, температура, активность сердца и других. Беспроводные имплантируемые датчики – это сложные биомедицинские микроэлектромеханические системы (БиоМЭМС), включающие биомедицинские диагностические и лечебные устройства с интегрированными микроэлектронными и микромеханическими компонентами. К беспроводным имплантируемым приборам предъявляются жесткие требования в части минимизации габаритов, обеспечения длительной работоспособности, биосовместимости и т.д. Технологии создания таких датчиков – сложный междисциплинарный процесс, для решения которой необходимо комплексно решить ряд научных и инженерных задач в области микроэлектроники, радиотехнологий, материаловедения и медицины. Наиболее оптимальным для проведения исследований является использование специального технологического оборудования, имитирующего функционирование имплантируемых устройств в среде, которая по своим характеристикам (электродинамическим, механическим, и т.д.) аналогична реальным различным частям и/или тканям человеческого тела, позволяющего сократить сроки проведения лабораторных испытаний и доклинических исследований. В связи с этим в данной статье рассматриваются вопросы создания автоматизированной системы для исследований медицинских имплантируемых приборов, позволяющей адаптировать систему под различные сценарии испытаний и обеспечить хранение и обработку большого количества данных.

Структурная схема системы и организация обмена информацией. Структурная схема автоматизированной системы для проведения исследований медицинских имплантируемых приборов включает блок управления и пять тестовых ячеек, объединённых общей магистральной линией связи (рисунок 1). Блок управления выступает в роли «ведущего» (*master*-устройства), тогда как каждая тестовая ячейка функционирует как «ведомое» (*slave*-устройство), обмениваясь данными по стандартному протоколу *Modbus*. В качестве аппаратной основы для передачи данных выбран двухпроводный дифференциальный интерфейс *RS-485 (TIA/EIA-485)*, обеспечивающий полудуплексную связь и высокую помехоустойчивость в условиях лабораторных и производственных испытаний. Используемая топология позволяет минимизировать количество проводных линий и упростить монтаж: для всех устройств используются две сигнальные жилы (А и В), к которым параллельно подключаются приёмопередатчики интерфейса *RS-485*. Поддержка стандартных скоростей и возможность применения внешних приёмопередатчиков, совместимых с популярными микроконтроллерами и микрокомпьютерами (*Arduino*, *ESP32*, *Raspberry Pi CM4* и другие), делают решение масштабируемым и легко адаптируемым к изменениям состава оборудования.



Рисунок 1. Структурная схема автоматизированной система для проведения исследований медицинских имплантируемых приборов

Обмен данными регулируется протоколом *Modbus (Modbus RTU)*, в рамках которого блок управления последовательно обращается к каждой ячейке по её уникальному адресу (от 1 до 247). Формат послыки предусматривает передачу адреса, кода выполняемой функции и набора регистров с данными, а также контрольной суммы (*CRC*). Стандартные функции протокола *Modbus* (чтение и запись регистров, чтение дискретных входов/выходов, передача диагностической информации) обеспечивают универсальность системы.

Каждая тестовая ячейка содержит микроконтроллер, обрабатывающий запросы от блока управления и возвращающий ответные сообщения. Внутри тестовой ячейки размещаются исполнительные и измерительные устройства, среди которых: датчики температуры, насосы и клапаны для перекачки и регулирования состава буферного раствора, УФ-модули для облучения исследуемых материалов, нагреватели с системами *PID*-регулирования, дополнительные датчики. Все устройства внутри ячейки управляются локальным микроконтроллером, который, в свою очередь, обменивается данными с блоком управления через регистры *Modbus*.

Блок управления, представленный в виде микрокомпьютера (*Raspberry Pi CM4*) с установленным программным обеспечением, обеспечивает визуальный интерфейс пользователя, хранение данных в базе (*MariaDB*) и взаимодействие по сети *Ethernet/Wi-Fi* при необходимости. Программный модуль, реализующий протокол *Modbus*, опрашивает каждую ячейку по очереди, формирует запросы на чтение или запись регистров и обрабатывает ответы. Вся собранная информация сохраняется в базу данных и отображается оператору в реальном времени.

Программное обеспечение блока управления. Программное обеспечение (ПО) блока управления разработано в соответствии с требованиями ГОСТ ИЕС 62304-2022 «Изделия медицинские. Программное обеспечение. Процессы жизненного цикла» и обеспечивает централизованное управление пятью тестовыми ячейками через протокол *Modbus*, используя микрокомпьютер *Raspberry Pi CM4* в качестве аппаратной платформы. Применение *Raspberry Pi* даёт возможность функционировать под управлением операционной системы семейства Linux, а также использовать язык программирования *Python* с фреймворком *PyQt5*. Хранение данных организовано в базе *MariaDB*, что способствует надёжной и удобной интеграции с компонентами программного обеспечения. ПО выполняет функции контроля и диагностики, выявляя потенциальные отказы, неисправности и аварийные ситуации в блоке управления и тестовых ячейках. Способность системы обнаруживать сбои и формировать сигналы оповещения минимизирует риск нештатных ситуаций и обеспечивает безопасность работы оператора со стендом. ПО организовано по модульному принципу, что способствует расширяемости и прозрачному разделению функционала. Основные модули: *Core* (ядро программы), *Database* (работа с базой данных), *GUI* (графический интерфейс пользователя) и *Settings* (настройки). Диаграмма, отображающая модули ПО блока управления и их взаимодействие отображена на рисунке 2.

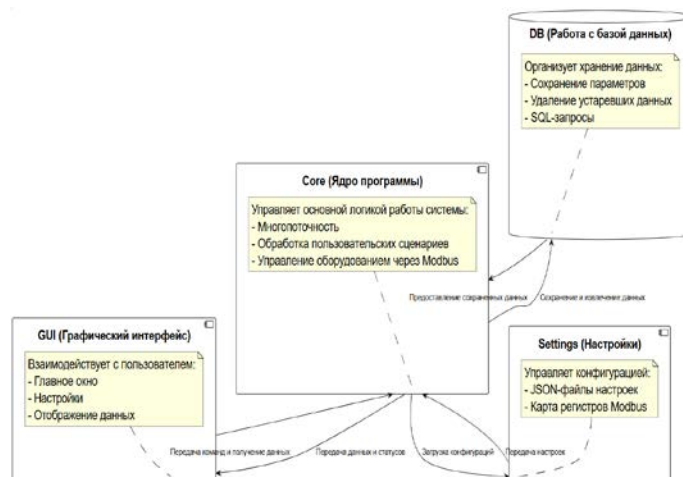


Рисунок 2. Диаграмма, отображающая модули ПО блока управления и их взаимодействие

Модуль *Core* обеспечивает многопоточную архитектуру и выполняет главную логику управления стендом. Он обрабатывает запросы от пользовательского интерфейса и взаимодействует с оборудованием посредством протокола *Modbus*. Данный модуль осуществляет координацию потоков данных, поддерживает синхронизацию процессов и распределяет задания между различными подсистемами. Параллельная обработка входящей/выходящей информации и результатов опроса датчиков даёт возможность оперативно реагировать на изменения параметров исследований, повышая общую производительность системы и уменьшая задержки.

Модуль *Database* организует хранение и доступ к данным, используя СУБД *MariaDB*, совместимую с архитектурой *ARM*. Структура базы ориентирована на высокую надёжность и удобство масштабирования. Модуль изолирован от прочих компонентов ПО, что позволяет вносить изменения в логику хранения или мигрировать на иную СУБД без нарушения функционирования других частей приложения. *Database* ведёт журнал изменений конфигурации, сохраняет результаты данных исследований, управляет удалением устаревших записей и даёт возможность оперативного доступа к собранным данным для анализа.

Модуль графического интерфейса пользователя (*GUI*) отвечает за взаимодействие с оператором, формируя удобное рабочее пространство. Он содержит набор окон и виджетов, позволяющих:

- управлять параметрами системой для проведения исследований;
- визуализировать данные в реальном времени в виде таблицы или графика;
- задавать сценарии автоматизированных исследований (последовательность команд и временные интервалы их исполнения).

Модуль *GUI* разработан на базе *PyQt5* и работает в многопоточном режиме, получая информацию от *Core* и отображая её в понятном для пользователя виде. Интерфейс выполнен с учётом требований ГОСТ ISO 10993-1–2021 относительно корректной настройки параметров УФ-воздействия на объекты, помещаемые в фантомы.

Модуль *Settings* отвечает за конфигурационные файлы в формате *JSON* (*JavaScript Object Notation*), где зафиксированы основные параметры работы ячеек: значения регистров *Modbus*, интервалы опроса, настройки нагревателей, конфигурация насосов и УФ-модулей. Данный подход с использованием *JSON* обеспечивает удобочитаемость настроек и гибкость при обновлениях: внесение изменений не требует правки исходного кода ПО. Модуль загружает необходимые конфигурации при запуске системы, а также предоставляет возможность их оперативного обновления на этапе исследовательских работ без перезапуска ПО. Информация из *Settings* передаётся в *Core*, а при необходимости сохраняется в *Database*.

Все модули объединены в единую систему обмена данными, где *Core* выступает ключевым связующим звеном. *Core* взаимодействует с *Database* для записи результатов и считывания архива, а также получает настройки через *Settings*. Данные, необходимые пользователю, отображаются в *GUI*, а команды, заданные оператором или сценариями исследований, передаются обратно в *Core*, который транслирует их оборудованию по *Modbus*. Данная схема исключает дублирование логики и упрощает обслуживание.

ПО предназначено для работы в круглосуточном режиме, что предусматривает отработку исключительных ситуаций: внезапные сбои сетевого взаимодействия, непредвиденные перепады напряжения, отключение отдельных тестовых ячеек. Блок управления формирует аварийные сообщения при обнаружении нештатных событий, регистрирует их в базе и выводит на экран оператора. Модульная структура и соответствие современным стандартам делают систему масштабируемой, позволяя расширять функционал или подключать новые модули без изменения базовых принципов работы.

Сценарии проведения исследований. Сценарии проведения исследований представляют собой детально прописанную последовательность действий, связанных с изменением параметров в одном или нескольких стендах, чтобы обеспечить максимально точное воспроизведение условий испытаний для медицинских имплантируемых приборов. Его подготовка начинается во внешнем приложении, в котором оператор указывает целевые значения для различных параметров (скорость вращения рабочего насоса, температура нагревательного модуля, длительность или частота импульсов ультрафиолетового облучения), а также время, через которое каждое действие должно быть выполнено относительно начала эксперимента.

После завершения подготовки сценарий сохраняется в специальный файл, который загружается в основное программное обеспечение блока управления. Программа анализирует структуру сценария, определяет его общую продолжительность и отображает основные сведения на экране оператора. В частности, пользователь может видеть, сколько времени займёт выполнение всех действий, на какие стенды направлены соответствующие команды и как именно изменятся параметры. При необходимости оператор запускает сценарий, и система пошагово отрабатывает запланированные изменения. Если для некоторых экспериментов нужно вводить разные условия в разных стендах, сценарий допускает включение нескольких параллельных ветвей. Например, в одной ячейке автоматически поднимается температура, в другой поддерживается стабильный нагрев и регулируется интенсивность ультрафиолетового облучения, а в третьей меняется скорость подачи буферной жидкости. В процессе выполнения каждый шаг сценария сопровождается записью фактических значений в базу данных: в ней фиксируются временные метки, состояние насосов и клапанов, текущая температура или мощность УФ-излучения. Когда очередной шаг срабатывает в момент, заданный пользователем, программа проверяет корректность передачи команд, отслеживает их выполнение, а также реагирует на возможные сбои. При возникновении нештатных обстоятельств (выхода за пределы заданной температуры или выявления неисправности в одном из узлов) оператор имеет возможность приостановить ход сценария, устранить проблемы и продолжить работу с момента прерывания.

При многоячейковой конфигурации сценарии могут варьировать условия испытаний для каждого стенда отдельно, что особенно востребовано при сравнительных тестах различных типов имплантируемых приборов. Система способна параллельно обрабатывать несколько потоков команд, не ограничиваясь единым набором параметров, что обеспечивает высокую гибкость и масштабируемость исследований. Пользователь, запустив сценарий, видит ход выполнения в режиме реального времени и может проанализировать, какие действия уже завершены и какие ещё предстоит выполнить. В случае необходимости изменения плана существует возможность откорректировать значения или интервалы до завершения экспериментального цикла.

Сценарии проведения исследований предоставляют операторам возможность комплексно и гибко управлять параметрами испытаний, формировать многоуровневые планы автоматического изменения среды в каждой ячейке и добиваться высокой точности в воспроизведении экспериментальных условий. Параллельное выполнение команд по нескольким стендам сокращает время для интегральных исследований, а система мониторинга и логирования даёт возможность приостанавливать и возобновлять сценарий при сохранении всех ранее собранных данных. Подобная организация экспериментов открывает широкие перспективы для исследователей, разрабатывающих и тестирующих медицинские имплантируемые приборы различной сложности.

Заключение. Представлена автоматизированная система для проведения исследований медицинских имплантируемых приборов. Система обеспечивает централизованное управление пяти и более тестовыми ячейками в соответствии с заданным оператором сценарием проведения исследований. Описаны архитектура программного обеспечения, включающего взаимодействие с аппаратной платформой по протоколу *Modbus*, управление датчиками и исполнительными устройствами, а также способ хранения и обработки данных с использованием СУБД. Модульная структура программного обеспечения и использование современных инструментов позволяют адаптировать систему под различные сценарии испытаний имплантируемых датчиков различных типов, обеспечить хранение и обработку большого количества данных исследований. Использование автоматизированной системы на практике позволит сократить время проведения исследований.

Список литературы

- [1] A miniature implantable RF-wireless active glaucoma intraocular pressure monitor : научная статья / E. Y. Chow [и др.] – IEEE Trans. Biomed : Circuits Syst, 2010. – 350 с.
[2] Программно-аппаратный комплекс ускоренных испытаний имплантированных медицинских датчиков : научная статья / А.Н. Осипов [и др.] – Минск : ResearchGate, 2025. – 5 с.
[3] WaveShare [Электронный ресурс]. – Режим доступа : <https://www.waveshare.com/wiki/CM4-DISPBASE-7A-BOX>.

Авторский вклад

Авторы внесли равный вклад в написание статьи

AUTOMATED SYSTEM FOR CONDUCTING RESEARCH ON MEDICAL IMPLANTABLE DEVICES

A.N. Osipov

*Head of the Research Laboratory,
BSUIR, PhD of Technical
Sciences, Associate Professor*

S.P. Urbanovich

*The leading engineer of the
Research Laboratory, BSUIR*

M.P. Batura

*Head of the Research Laboratory,
BSUIR, Doctor of Technical Sciences,
Professor*

A.A. Ivanovsky

Student, BSUIR

I.O. Khazanovsky

Postgraduate student, BSUIR

A.V. Korotkevich

*Associate Professor of the
Department of Micronanoelectronics,
BSUIR, PhD of Technical Sciences,
Associate Professor*

Abstract. Approaches to the creation of an automated system for the research of medical implantable devices are considered. The architecture of the software is described, including interaction with the hardware platform via the Modbus protocol, control of sensors and actuators, as well as a method for storing and processing data using a DBMS (Database Management System). It is shown that the modular structure of the software and hardware complex and the use of modern tools make it possible to adapt the system to various test scenarios, to ensure the storage and processing of a large amount of research data.

Keywords: automated system, medical implantable devices, accelerated testing, Modbus protocol, Raspberry Pi CM4, PyQt5, electromagnetic compatibility, control of experimental parameters.