

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

**РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ СБОРА,  
АГРЕГАЦИИ И АНАЛИЗА ДАННЫХ С GSM-ТЕРМИНАЛОВ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы  
направления 38.03.05 — Бизнес-информатика

механико-математического факультета  
Абрамова Серафима Андреевича

Научный руководитель  
доцент, к. ф.-м. н.

\_\_\_\_\_

Д. В. Мельничук

Заведующий кафедрой  
д. ф.-м. н., доцент

\_\_\_\_\_

С. П. Сидоров

Саратов 2026

**Введение.** В условиях цифровизации электроэнергетической отрасли возрастают требования к надёжности и оперативности мониторинга промышленного оборудования. Устройства УОПИ (устройства обработки и передачи информации), установленные на электрических подстанциях, генерируют SMS-уведомления при авариях, отключениях и изменениях статуса оборудования. Однако традиционные методы обработки таких сообщений, такие как ручная проверка телефона модема, не обеспечивают централизованного хранения, анализа и оперативного доступа к данным. Это создаёт существенные барьеры для эффективного реагирования на инциденты и ведения ретроспективного анализа событий.

В качестве ответа на эти вызовы в работе предлагается автоматизированная система мониторинга SMS-уведомлений, способная собирать, фильтровать, хранить и отображать сообщения от промышленных устройств в реальном времени через веб-интерфейс. Разработанное решение устраняет ключевые недостатки ручных методов обработки: сообщения больше не теряются в памяти модема, появляется полная история событий с возможностью поиска и фильтрации, формируется статистика по подстанциям и устройствам, а участие оператора сводится к минимуму благодаря автоматическому обновлению интерфейса.

**Целью данной бакалаврской работы** является разработка программного комплекса для автоматизированного сбора и мониторинга SMS-уведомлений от промышленных устройств УОПИ на электрических подстанциях с интеграцией веб-интерфейса для операторов. Достижение этой цели подразумевает создание полнофункциональной системы, работающей в автономном режиме 24/7, способной обрабатывать до 1500 сообщений в сутки с задержкой отображения не более 40 секунд от момента приёма до момента отображения в веб-интерфейсе.

Для достижения поставленной цели в работе были сформулированы и решены следующие **задачи**:

- Провести анализ существующих решений в области мониторинга промышленного оборудования и выявить их ограничения;
- Исследовать методы работы с GSM-модемами через AT-команды и формат PDU для SMS;

- Разработать архитектуру системы, включающую модули сбора SMS, фильтрации, хранения и веб-представления;
- Реализовать прототип системы на базе Python с использованием библиотек pycserial, Flask, SQLite;
- Продемонстрировать работу системы на реальных сценариях;
- Оценить перспективы внедрения и развития разработанного решения.

**Объектом исследования** выступают системы мониторинга промышленного оборудования с использованием GSM-коммуникации.

**Предметом исследования** является архитектура, методы и технологии разработки программных комплексов для сбора и обработки SMS-уведомлений от промышленных устройств.

**Научная новизна** данной работы заключается в предложении архитектуры системы мониторинга, объединяющей несколько ключевых компонентов. Система обеспечивает автоматический опрос GSM-модема через AT-команды с интервалом 30 секунд. Реализована конвертация PDU-формата SMS в читаемый текст с защитой от повторной обработки через уникальные хэши сообщений. Внедрена фильтрация сообщений по активным и неактивным номерам с хранением списков в таблице базы данных. Предоставлен веб-интерфейс реального времени с авто-обновлением каждые 10 секунд, использующий механизм cache-busting. Обеспечено архивирование базы данных перед каждым запуском системы.

**Практическая значимость** заключается в возможности внедрения системы на электрических подстанциях для повышения оперативности реагирования на аварии, ведения истории событий и формирования статистики по оборудованию. Решение может быть использовано как на одиночных объектах, так и масштабировано на сеть подстанций за счёт модульной архитектуры и низких требований к инфраструктуре. Система не требует выделенного сервера баз данных, работает на стандартном компьютере с операционной системой Windows или Linux, использует доступное оборудование в виде GSM-модема с USB-подключением.

В ходе выполнения работы использовался комплекс методов. Методы анализа предметной области включали изучение современных систем мониторинга промышленного оборудования. Методы архитектурного проектирования

применялись для построения модульной структуры системы. Методы работы с последовательным портом реализованы через библиотеку `pyserial` для отправки AT-команд. Методы веб-разработки включают использование фреймворка `Flask` для создания REST API. Методы работы с базами данных охватывают проектирование схемы `SQLite`. Инструментальные средства включают `Python 3.x`.

**В первой главе** проводится анализ современного состояния систем мониторинга промышленного оборудования. Рассмотрены предпосылки к созданию автоматизированных систем сбора SMS-уведомлений, а также существующие решения.

Глава начинается с обзора эволюции систем диспетчеризации в электроэнергетике с 2010 по 2025 год. Отмечается рост числа устройств УОПИ на подстанциях и увеличение объёма событий. Ручная проверка телефона модема стала неэффективной, что привело к необходимости автоматизации сбора уведомлений.

Раскрываются проблемы операторов при ручной обработке SMS. Сообщения теряются в памяти модема из-за ограниченного объёма хранения. Отсутствует история и возможность ретроспективного анализа. Нет статистики по подстанциям и устройствам. Требуется постоянное участие оператора для проверки телефона.

Проанализирована роль GSM-коммуникации в промышленном мониторинге. Для многих удалённых подстанций GSM остаётся единственным доступным каналом связи. Важно обеспечить устойчивость к перебоям связи и сохранение данных при сбоях питания.

Особое внимание уделено обзору и критическому сравнению существующих решений. Проприетарные системы диспетчеризации обладают мощным функционалом, но отличаются высокой стоимостью и сложностью развёртывания. SCADA-системы предоставляют богатую аналитику, однако требуют специализированного оборудования и протоколов, таких как `Modbus` и `IEC 61850`. Готовые SMS-шлюзы обеспечивают простую интеграцию, но имеют ограниченный функционал фильтрации и хранения. Самописные решения на `Python` отличаются гибкостью и низкой стоимостью, но требуют квалификации для поддержки.

На основе обзора делается вывод, что ни одно из существующих решений не обеспечивает оптимального баланса между стоимостью, функциональностью и простотой развёртывания для небольших подстанций. Обосновывается потребность в новой системе мониторинга, которая сочетает простоту развёртывания, низкую стоимость оборудования и достаточный функционал.

**Вторая глава** бакалаврской работы посвящена раскрытию теоретических основ, необходимых для обоснования архитектуры и компонентов разрабатываемой системы. Основное внимание уделяется методам работы с GSM-модемами, форматам SMS-сообщений и архитектурным паттернам веб-приложений. Рассмотрены как технические детали реализации, так и ограничения, с которыми сталкивается разработчик при работе с оборудованием.

В начале главы даётся описание принципов работы GSM-модемов. Подключение осуществляется через последовательный порт, эмулируемый через USB-интерфейс, что обеспечивает совместимость с современными компьютерами без физических COM-портов. Управление производится через AT-команды, стандартизированные набором Hayes command set, который поддерживается большинством производителей модемов. Режимы работы с SMS включают текстовый режим, удобный для отладки, и PDU-режим, необходимый для корректной обработки кириллических сообщений. Хранение сообщений осуществляется в памяти модема, доступной через обозначения ME для памяти устройства и SM для памяти SIM-карты.

Рассматриваются основные AT-команды для работы с SMS:

- команда AT+CMGF выбирает режим работы с сообщениями, переключая между текстовым и PDU-форматом;
- команда AT+CMGL читает список сообщений из памяти модема, возвращая заголовки и тела сообщений в формате, требующем дальнейшего парсинга;
- команда AT+CMGD удаляет сообщения из памяти после успешной обработки, что предотвращает повторное чтение;
- команда AT+CSQ запрашивает уровень сигнала, позволяя оценивать качество связи в реальном времени;
- команды AT+CGMI, AT+CGMM и AT+CGMR возвращают информацию о производителе, модели и ревизии модема, что используется для

мониторинга статуса оборудования.

Во второй части главы рассматриваются особенности формата PDU. Структура PDU-сообщения включает адрес отправителя, временную метку SCTS, заголовок пользовательских данных UDH для составных сообщений и сами данные в кодировке UCS2 или GSM 7-bit. Кодировки текста требуют особого внимания: GSM 7-bit эффективна для латиницы, но UCS2 обязательна для кириллических сообщений. Конкатенация составных SMS реализуется через UDH с reference ID, что позволяет собирать сообщения из нескольких частей в правильном порядке. Алгоритмы конвертации HEX в текст включают декодирование байтов, обработку порядка байт для UTF-16 и фильтрацию служебных данных.

Кроме того, обсуждаются известные ограничения при работе с GSM-модемами. Задержки при опросе модема возникают из-за последовательного характера обмена данными и необходимости ожидания ответов на каждую команду. Возможность потери сообщений при сбоях питания требует реализации механизмов архивирования и восстановления данных. Смешивание служебных уведомлений, таких как +СМТІ о новых сообщениях, с ответами на команды требует фильтрации при парсинге ответов. Необходимость фильтрации дубликатов решается через вычисление хэшей сообщений и проверку уникальности перед сохранением в базу данных.

В завершение главы проводится обоснование выбора архитектурных подходов и технологий:

- Python — скорость разработки, богатая экосистема библиотек;
- SQLite — не требует отдельного сервера, достаточно одного файла;
- Flask — минималистичный веб-фреймворк для внутреннего использования;
- Модульная архитектура — разделение сборщика, базы и веб-сервера.

**В третьей главе** описана архитектура системы мониторинга SMS, представлен выбор технологий и приведено описание реализации.

Сформулированы требования к функциональности системы. Автоматический опрос GSM-модема осуществляется с интервалом 30 секунд. Конвертация PDU-формата включает защиту от повторной обработки. Фильтрация по активным и неактивным номерам реализована через таблицу в базе данных.

Хранение сообщений обеспечивает защиту от дубликатов через уникальные хэши. Веб-интерфейс с авто-обновлением каждые 10 секунд использует cache-busting. Мониторинг статуса модема осуществляется через периодический запрос AT-команд. Архивирование базы данных создаёт резервную копию с временной меткой.

Предложена модульная архитектура, включающая следующие компоненты:

1. **Сборщик SMS (`gsm_sms_collector.py`)** — опрашивает модем каждые 30 секунд, извлекает сообщения, конвертирует PDU;
2. **База данных (SQLite)** — хранение сообщений с хэшами, временными метками, привязкой к подстанции и устройству;
3. **Веб-сервер (`web_server.py`)** — Flask API для получения данных, раздача статических файлов;
4. **Веб-интерфейс (HTML/CSS/JS)** — отображение сообщений, статистики, статуса модема;
5. **Утилиты (`sms_utils.py`)** — конвертация PDU, нормализация номеров, сборка составных SMS;
6. **Лаунчер (`launcher.py`)** — запускает процессы в отдельных потоках, обрабатывает Ctrl+C.

Показано взаимодействие с внешними сервисами: GSM-модем принимает SMS через AT-команды, JSON-файл `modem_status.json` обеспечивает обмен статусом между компонентами.

Уделено внимание сценариям обработки ошибок. Автоматическое переподключение при потере связи реализуется через цикл повторных попыток с интервалом 10 секунд. Логирование в файле `gsm_modem.log` позволяет анализировать работу системы. Архивирование БД защищает от потери данных. Защита от повторной конвертации PDU достигается проверкой формата данных.

Реализованный прототип обеспечивает приём и обработку до 1500 SMS в сутки, фильтрацию по активным и неактивным номерам, отображение статистики за 24 часа, мониторинг статуса модема в реальном времени.

**Четвёртая глава** работы посвящена практической демонстрации возможностей разработанной системы. На основе реализованной архитектуры и

функциональных модулей представлены ключевые сценарии использования, иллюстрирующие работу системы в реальных условиях эксплуатации. Каждый сценарий сопровождается подробным описанием происходящих процессов и результатами.

Глава начинается с описания процесса запуска системы. Запуск через файл `start.bat` включает проверку зависимостей, доступности портов и наличия необходимых файлов конфигурации. После успешной проверки автоматически открывается браузер на адресе `http://localhost:5000`, где отображается главная страница веб-интерфейса. На странице отображается статус модема с указанием модели и производителя, статистика по сообщениям за всё время и за последние 24 часа, а также таблица последних сообщений с возможностью поиска и фильтрации.

Затем приводится серия кейсов, раскрывающих реальные возможности системы.

Кейс 1 демонстрирует приём SMS от устройства УОПИ: устройство отправляет SMS при изменении статуса, модем принимает сообщение, сборщик опрашивает модем через 30 секунд, конвертирует PDU, сохраняет в базу, и оператор видит сообщение в веб-интерфейсе через 10 секунд после сохранения. Общая задержка от приёма до отображения составляет от 10 до 40 секунд в зависимости от момента прихода сообщения относительно цикла опроса.

Кейс 2 показывает фильтрацию по активным номерам: сообщение от неизвестного номера не сохраняется в базу и удаляется из модема, сообщение от номера из таблицы активных номеров сохраняется с привязкой к подстанции и устройству, сообщение от неактивного номера сохраняется в базе, но не отображается в веб-интерфейсе.

Кейс 3 иллюстрирует обработку отключения модема: при физическом отключении модема статус в веб-интерфейсе меняется на «Отключен» в течение 15-30 секунд, после повторного подключения модем автоматически переподключается без вмешательства оператора, история сообщений сохраняется в базе данных.

Кейс 4 демонстрирует авто-обновление интерфейса: новые сообщения появляются в таблице без ручной перезагрузки страницы благодаря `cache-busting` через уникальный `timestamp` в API-запросах, позиция прокрутки

таблицы сохраняется при обновлении, что позволяет оператору продолжать просмотр без потери контекста.

Кейс 5 показывает архивирование базы данных: перед каждым запуском создаётся копия БД в папке `archive_db/` с временной меткой в формате `ДД_ММ_ГГГГ_ЧЧ_ММ_СС`, при сбоях возможно восстановление данных из последней резервной копии.

Кейс 6 демонстрирует механизм оперативного поиска и фильтрации сообщений в веб-интерфейсе. При вводе ключевого слова, номера телефона или названия подстанции клиентская часть формирует асинхронный GET-запрос к эндпоинту `/api/search`, который выполняет полнотекстовый поиск по полям `message_conv`, `phone_number` и `substation` с использованием оператора `LIKE`. Результаты ограничены 50 записями и отображаются без перезагрузки страницы. Дополнительно реализован фильтр по подстанциям через выпадающий список, позволяющий сузить выборку до событий конкретного объекта без выполнения текстового поиска.

Каждый из сценариев сопровождается скриншотами реального интерфейса и подробным пояснением происходящих процессов. Это позволяет оценить пользовательский опыт и подтвердить выполнение всех заявленных функций. Скриншоты включают главную страницу со статистикой, таблицу сообщений с фильтрацией, модальное окно статуса модема и примеры лог-файлов с записями о событиях системы.

Итоговые метрики системы подтверждают достижение поставленных целей:

- Производительность: 1500 SMS в сутки в зависимости от количества подключённых устройств;
- Задержка отображения: от 10 до 40 секунд от приёма до веб-интерфейса;
- Надёжность: автоматическое переподключение при сбоях связи с модемом;
- Хранение: неограниченная история в SQLite с архивированием;
- Платформа: Windows 10/11 или Linux.

**Заключение.** В рамках данной бакалаврской работы была решена задача проектирования и реализации системы мониторинга SMS-уведомлений от промышленных устройств УОПИ на электрических подстанциях. Выполненное

исследование охватывало как теоретические аспекты работы с GSM-модемами и форматами SMS, так и прикладную реализацию полнофункционального прототипа. Разработанная система подтвердила возможность создания надёжного инструмента мониторинга, сочетающего простоту развёртывания и достаточный функционал для оперативной работы оператора.

По итогам работы были достигнуты следующие **результаты**:

- Выполнен анализ существующих решений в области мониторинга промышленного оборудования, выявлены их ограничения и потребность в более доступном и простом инструменте для небольших подстанций;
- Обоснованы архитектурные принципы и технологии построения системы мониторинга SMS, рассмотрены методы работы с AT-командами и PDU-форматом, включая особенности конвертации и обработки составных сообщений;
- Предложена и реализована модульная архитектура системы с веб-интерфейсом, способная автоматически собирать, фильтровать и отображать SMS-уведомления в реальном времени с задержкой не более 40 секунд;
- Разработан прототип на языке Python с использованием библиотек puserial, Flask, openpyxl, SQLite, реализующий базовые сценарии: приём SMS, конвертация PDU, фильтрация по активным и неактивным номерам, веб-отображение с авто-обновлением;
- Выполнена демонстрация работы системы с визуализацией интерфейса и результатами обработки реальных SMS от устройств УОПИ, подтверждена работоспособность всех заявленных функций.

Разработанная система подтвердила возможность создания надёжного инструмента мониторинга, сочетающего простоту развёртывания и достаточный функционал для оперативной работы оператора. Работа создаёт основу для дальнейшего расширения системы — в том числе за счёт интеграции с SCADA-системами и масштабирования на сеть подстанций. Модульная архитектура позволяет добавлять новые компоненты без переписывания существующего кода, что обеспечивает долгосрочную поддерживаемость решения.

**Планы на будущее:**

- Экспорт отчётов в Excel и PDF по периоду;
- Авторизация пользователей в веб-интерфейсе;

- Поддержка нескольких GSM-модемов для отказоустойчивости;
- Интеграция с диспетчерской SCADA-системой.

Таким образом, поставленные цели и задачи исследования были выполнены в полном объёме, а результаты обладают высокой практической значимостью для сферы мониторинга промышленного оборудования в электроэнергетике.