Акционерное общество «Энфорс»
ИНН 9723248779 КПП 772301001 ОГРН 1257700080659
109380, г. Москва, вн.тер.г. Муниципальный округ Люблино,
ул Чагинская, дом 4, строение 13, помещение 14/4

Инженерное превосходство в каждом ватте
info@enforce.moscow | enforce.moscow

Утверждаю
Генеральный директор АО
«ЭНФОРС»
Гуреев Михаил Владимирович
_____/
«30» апреля 2026 г.

Программное обеспечение «Интеллектуальная система управления системой накопления и хранения электрической энергии на основе батарей аккумуляторных литий-ионных (тяговой батареей)»

ОПИСАНИЕ ТЕХНИЧЕСКОЙ АРХИТЕКТУРЫ

Документация на программное обеспечение

Автор: Гуреев Михаил Владимирович

г. Москва, 2026

Акционерное общество «Энфорс»
ИНН 9723248779 КПП 772301001 ОГРН 1257700080659
109380, г. Москва, вн.тер.г. Муниципальный округ Люблино,
ул Чагинская, дом 4, строение 13, помещение 14/4

СОДЕРЖАНИЕ

Акционерное общество «Энфорс»
ИНН 9723248779 КПП 772301001 ОГРН 1257700080659
109380, г. Москва, вн.тер.г. Муниципальный округ Люблино,
ул Чагинская, дом 4, строение 13, помещение 14/4

1 Общие сведения

Настоящий документ содержит описание технической архитектуры программы для ЭВМ «Интеллектуальная система управления системой накопления и хранения электрической энергии на основе батарей аккумуляторных литий-ионных (тяговой батареей)». Документ подготовлен для представления в составе комплекта документации на программное обеспечение и предназначен для экспертной оценки состава, назначения, модульной структуры и технологических решений экземпляра программы.

Правообладателем программы является Акционерное общество «ЭНФОРС». Автор программы: Гуреев Михаил Владимирович. Согласно реферату, программа относится к программному обеспечению, реализованному на языках C/C++ и предназначенному для работы в средах Windows и Linux при экспертной проверке, а также для применения в составе программно-аппаратного комплекса управления батареей.

Программа представляет собой многопоточную операционную систему реального времени и набор сервисных программных модулей для микроконтроллеров на ядре ARM. Архитектура ориентирована на управление задачами, синхронизацию сервисов, распределение ресурсов, контроль периферии, работу с файловой структурой, сетевыми протоколами и вычислениями с плавающей точкой.

2 Назначение и границы архитектуры

Архитектура описывает программные компоненты, обеспечивающие непрерывный мониторинг параметров элементов батареи, балансировку ячеек, защиту от опасных режимов, расчет ключевых показателей состояния батареи, управление зарядом и разрядом, термоконтроль, ведение журнала данных и обмен информацией с внешними системами.

Границы рассматриваемой архитектуры включают программные модули экземпляра ПО, его абстракции аппаратной платформы, подсистемы сборки, файлового ввода-вывода, сетевого и сервисного обмена, а также точки взаимодействия с датчиками, исполнительными устройствами и внешней системой управления. Аппаратная часть батареи, силовые цепи и конкретные конструктивные параметры изделия рассматриваются как внешняя среда по отношению к программе.

- входными данными являются измерения напряжения, тока, температуры, диагностические признаки и конфигурационные параметры;
- выходными данными являются управляющие воздействия, сообщения диагностики, журнальные записи и сервисные данные для внешних систем;
- управляющая логика реализуется в виде набора независимых сервисов, выполняемых под управлением планировщика задач;
- платформенная независимость обеспечивается через слои абстракции операционной системы, файловой системы и компилятор-специфичных определений.

3 Контекстная архитектура

На контекстном уровне программа располагается между физическими компонентами батареи и внешними системами мониторинга либо управления. Через нижний уровень она получает измерения от датчиков и формирует команды исполнительным устройствам. Через верхний уровень она передает диагностическую и эксплуатационную информацию, включая данные по шине CAN 2.0b.

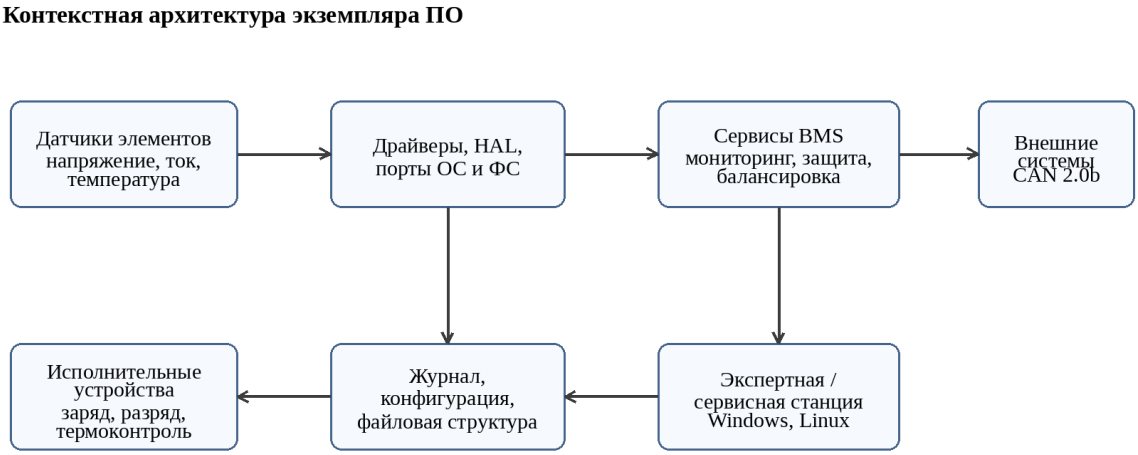

Рисунок 1 - Контекстная архитектура экземпляра ПО

Внешняя сервисная или экспертная станция применяется для проверки экземпляра ПО, анализа журналов, загрузки конфигурации, контроля корректности обмена и фиксации результатов экспертной проверки. В составе архива присутствуют материалы, указывающие на использование переносимых портов операционной системы, файловой системы, сетевого стека и сборочной конфигурации.

4 Логическая структура программных модулей

Логическая структура программы построена по многослойному принципу. Верхний уровень реализует прикладные функции управления батареей. Средний уровень содержит сервисные и инфраструктурные подсистемы. Нижний уровень обеспечивает переносимость между аппаратными платформами, операционными окружениями и файловыми системами.

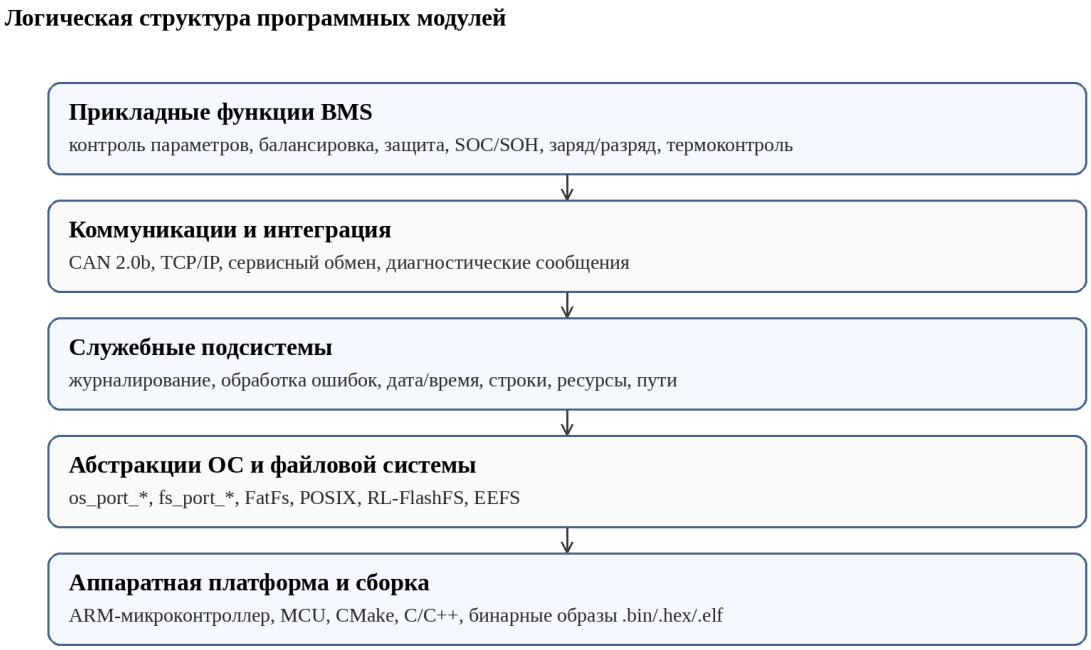

Рисунок 2 - Логическая структура программных модулей

Такое разделение упрощает перенос программы между целевым контроллером и стендом экспертной проверки, снижает связанность модулей и позволяет изолированно тестировать сервисы, отвечающие за контроль, диагностику и обмен данными.

5 Взаимодействие модулей и поток данных

Типовой цикл обработки начинается со сбора телеметрии с элементов батареи. После получения данные нормализуются, проверяются на допустимые диапазоны и передаются в диагностические сервисы. По результатам диагностики прикладные сервисы формируют управляющие решения: включение или отключение зарядных и разрядных режимов, запуск балансировки, активация термоконтроля, запись события в журнал и отправка сообщения во внешнюю систему.

Акционерное общество «Энфорс»
ИНН 9723248779 КПП 772301001 ОГРН 1257700080659
109380, г. Москва, вн.тер.г. Муниципальный округ Люблино,
ул Чагинская, дом 4, строение 13, помещение 14/4

Поток обработки данных и управляющих воздействий

Рисунок 3 - Поток обработки данных и управляющих воздействий

Таблица 1 - Этапы обработки данных и формирования управляющих воздействий

Этап	Описание	Результат
Сбор данных	периодический опрос датчиков напряжения, тока и температуры каждого элемента батареи	набор первичных измерений
Нормализация	фильтрация, проверка единиц измерения и допустимых диапазонов	подготовленные значения для логики управления
Диагностика	выявление перегрузки, короткого замыкания, перегрева, глубокого разряда и иных опасных режимов	диагностический статус и код события
Управление	выбор безопасного действия: балансировка, ограничение заряда, отключение нагрузки, включение охлаждения или подогрева	команда на исполнительные устройства
Журналирование и обмен	запись истории работы и передача сообщений во внешние системы	журнальная запись и пакет обмена

6 Сборочная и развертываемая конфигурация

Материалы архива показывают, что сборка экземпляра ПО организуется через CMake. В сборочную конфигурацию включаются модули MCU, Unilib, FatFs, EEFS, CycloneTCP и файловые списки прикладных исходников. После сборки формируются исполняемый файл и производные машинные образы, пригодные для загрузки или экспертного анализа.

- исходная конфигурация сборки задает минимальную версию CMake и перечень подключаемых модулей;
- в процессе сборки формируются сведения о дате и времени сборки, что необходимо для идентификации экземпляра;
- после сборки создаются артефакты в форматах .bin, .hex и .elf;
- наличие портов Windows, POSIX и ряда RTOS позволяет использовать переносимые интерфейсы в разных режимах проверки и исполнения.

7 Обоснование выбора технологий

Таблица 2 - Обоснование выбора технологий реализации программного обеспечения

Технология	Обоснование применения
C/C++	обеспечивает малые накладные расходы, прямой контроль памяти, предсказуемую производительность и пригодность для микроконтроллерных систем
ARM-ориентированная RTOS-архитектура	позволяет организовать детерминированное выполнение задач реального времени и синхронизацию независимых сервисов
CMake	унифицирует сборку и позволяет формировать воспроизводимые артефакты для разных окружений
Абстракции os_port_* и fs_port_*	снижают зависимость прикладной логики от конкретной операционной системы и файловой реализации
FatFs, EEFS, файловая структура	обеспечивают хранение конфигурации, журналов и диагностических данных во встроенной среде
TCP/IP и CAN 2.0b	обеспечивают обмен диагностической и эксплуатационной информацией с внешними системами

Выбранные технологии соответствуют назначению программы: управление батареей требует надежной работы в реальном времени, минимальных задержек

реакции на опасные режимы, переносимости и возможности последующей диагностики.

8 Надежность, безопасность и сопровождаемость

Надежность архитектуры обеспечивается разделением функций по модулям, контролем ошибок, журналированием событий, поддержкой безопасных состояний и использованием сервисов реального времени. В аварийных ситуациях программа должна приоритетно обеспечить отключение опасного режима, зафиксировать событие и передать диагностическую информацию.

- при выходе параметров за допустимые пределы выполняется переход в безопасное состояние;
- журнал работы сохраняет историю событий, ошибок и циклов батареи для последующего анализа;
- модульная структура упрощает сопровождение, тестирование и замену отдельных подсистем;
- сведения о версии и времени сборки позволяют однозначно идентифицировать проверяемый экземпляр;
- наличие переносимых портов упрощает экспертную проверку вне целевой аппаратной платформы.

Представленная архитектура является достаточной для описания технического устройства экземпляра ПО, его основных модулей, каналов взаимодействия и технологической базы, используемой для реализации функций управления батареей.