

Программное обеспечение системы автоматического управления электронасосным агрегатом

Описание процессов, обеспечивающих
поддержание жизненного цикла программного
обеспечения

ООО «РУ-ДРАЙВ ЦИФРОВЫЕ ТЕХНОЛОГИИ»

13.05.2026

Описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения

Оглавление

1	ВВЕДЕНИЕ	2
2	ОБЩИЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ	2
2.1	Принятые сокращения	2
2.2	Основные термины	2
3	ОПИСАНИЕ ПРОЦЕССОВ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	3
3.1	Заказ (приобретение) программного обеспечения	3
3.2	Поставка	3
3.3	Разработка	3
3.3.1	Анализ требований	4
3.3.2	Проектирование архитектуры	4
3.3.3	Программирование, сборка, тестирование, документирование	5
3.4	Эксплуатация	5
3.4.1	Эксплуатация	5
3.4.2	Техническая поддержка пользователей	6
3.4.3	Состав и квалификация персонала	7
4	РЕШЕНИЕ ТИПОВЫХ ПРОБЛЕМ	8
5	КОНТАКТНАЯ ИНФОРМАЦИЯ ПРОИЗВОДИТЕЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	9
5.1	Юридическая информация	9
5.2	Контактная информация службы технической поддержки	9

1 Введение

Программное обеспечение предназначено для реализации системы автоматического управления электронасосным агрегатом. Разработанный алгоритм обеспечивает контроль технологических параметров, предупредительную и аварийную сигнализацию, защитное отключение агрегата, световую сигнализацию состояний механизмов и запорных арматур. Программное обеспечение применяется для работы электронасосного агрегата.

2 Общие понятия и определения

2.1 Принятые сокращения

ПО – программное обеспечение

ПТК – программно-технический комплекс

ПЛК – программируемый логический контроллер

АРМ – автоматизированное рабочее место

ЧМИ (НМИ) – человеко-машинный интерфейс

2.2 Основные термины

Цех – совокупность производственных участков. Часть предприятия, где ведется производство. Может состоять из одного или нескольких технологических участков.

Технологический участок – место размещения или несколько мест размещения технологического оборудования, трубопроводов, арматуры и проч., выполняющие отдельную функцию как часть технологического процесса. Технологический участок может состоять из одного или нескольких технологических узлов.

Технологический узел – конструктивно и технологически обособленная часть объекта строительства, техническая готовность которой после завершения строительно-монтажных работ позволяет автономно, независимо от готовности объекта в целом проводить пусконаладочные работы, индивидуальные испытания и комплексное опробование агрегатов, механизмов и устройств.

Агрегат, объект - унифицированный механизм или узел механизма, выполняющий определенные функции. Под агрегатом (объектом) в данной системе понимаются её базовые составные части, имеющие возможность независимой работы от остальной системы и выполняющие команды оператора или САУ (такие как насос, задвижка, клапан, датчик, аналоговый выход), состояние которых можно наблюдать в режиме реального времени посредством АРМ.

Автоматизированное рабочее место (АРМ) - часть программно-технического комплекса (ПТК), предназначенная для автоматизации технологического процесса. АРМ объединяет программно-аппаратные средства, обеспечивающие взаимодействие человека с компьютером, панелью оператора (любым электронным устройством, обеспечивающим взаимодействие человека с объектом управления), предоставляет возможность ввода информации (через клавиатуру, компьютерную мышь, сканер и пр.) и её вывод на экран монитора, принтер, графопостроитель, звуковую карту, динамики или иные устройства вывода.

3 Описание процессов жизненного цикла программного обеспечения

Жизненный цикл ПО состоит из следующих этапов:

1. Заказ (приобретение); в рамках процесса выполняется формирование заявки на приобретение ПО со стороны заказчика, подготовка и корректировка договора, приемка.
 2. Поставка; производятся анализ и согласование требований, подготовка и подписание договора, выполнение работ и отгрузка ПО заказчику.
 3. Разработка; описывает процессы проектирования ПО.
 4. Эксплуатация; описывает процессы эксплуатации ПО пользователем и технической поддержки.
- Ниже описан каждый из этапов.

3.1 Заказ (приобретение) программного обеспечения

Возможны два способа приобретения ПО:

1. В составе разрабатываемого для заказчика программного обеспечения или программно-технического комплекса;
2. Как отдельного продукта по лицензионному договору.

Стоимость ПО рассчитывается индивидуально. Информация о стоимости ПО и о предоставлении права использования по лицензионному договору предоставляется по запросу заказчика (контактная информация приведена в п. 0).

3.2 Поставка

Процесс поставки ПО состоит из следующих этапов:

1. Анализ и согласование функциональных требований к системе (ПТК, программному комплексу, библиотеке) с заказчиком.
2. Подготовка и подписание договора с заказчиком:
 - на разработку и поставку программно-технического комплекса (системы), в составе которого используется ПО;
 - или на разработку и поставку программного обеспечения, в составе которого используется ПО;
 - или на поставку ПО как отдельного продукта.
3. Планирование и выполнение работ по поставке (в т.ч. разработка, см. п. 3.3).
4. Отгрузка заказчику разработанного программно-технического комплекса, программного обеспечения.

Если договором предусмотрена поставка ПО как отдельного продукта, то она поставляется на физическом носителе информации или в электронном виде в виде ссылки для скачивания через интернет.

3.3 Разработка

Процесс разработки библиотеки состоит из следующих этапов:

1. анализ требований;

2. проектирование архитектуры;
3. программирование;
4. сборка;
5. тестирование;
6. документирование.

При разработке ПО используется эволюционная модель, поэтому данные этапы периодически повторяются при изменении требований к ПО или возникновении новых.

Новый цикл разработки с корректировкой требований запускается в следующих случаях:

- требуется исправление ошибок в ПО; ошибка может быть выявлена как в результате тестирования, так и при эксплуатации;
- требуется добавление новых функций, функциональных блоков без изменения архитектуры ПО (расширение функционала) или изменение алгоритма работы существующих блоков;
- требуется внесение изменений в архитектуру ПО вследствие возникновения новых задач или применений библиотеки.

Ниже приведено описание отдельных этапов разработки.

3.3.1 Анализ требований

При анализе требований выполняются следующие работы:

1. Определяется отрасль промышленности/технология, для которой необходимо выполнить доработку (новую разработку) в ПО;
2. Выполняется поиск подходящего алгоритма (описания принципа работы) датчика, исполнительного механизма, технологического узла, технологического участка.
3. Определяется список минимально необходимых и достаточных изменений, которые нужно внести в ПО.
4. Проверяется выполнимость требуемых изменений;
5. Разрабатывается частное техническое задание (ЧТЗ) (или несколько) на доработку/разработку элементов ПО (в виде эскиза, чертежа, текстового документа, в устной форме).

3.3.2 Проектирование архитектуры

При проектировании архитектуры ПО выполняются следующие работы:

1. Разработка интерфейсов.

Под интерфейсом понимается упорядоченный набор данных, необходимых для обмена информацией с внутренней и внешней средой; интерфейсы делятся на два типа:

- внешний интерфейс; определяет структуры данных, предназначенные для обмена информацией с другими алгоритмами внутри программы ПЛК или с другими ПЛК, а также устройствами человеко-машинного интерфейса (HMI). В данном случае – это пользовательский тип данных (структура), который предназначен для вывода информации на устройства HMI, передачи команд пользователя от них в ПЛК, а также для обмена информацией с другими алгоритмами.
- внутренний интерфейс; определяет структуры данных, предназначенные для обмена информацией внутри программы ПЛК, в которой используется библиотека. К этим данным относятся входные/выходные параметры функций/функциональных блоков, а также их локальные переменные.

2. Разработка алгоритмов.

Описание принципа работы исполнительного механизма (технологического узла, технологической установки) адаптируется под поставленную задачу, и на ее основе разрабатывается алгоритм конкретной установки для ПО.

3. Разработка структур данных, необходимых для работы алгоритмов. К этим структурам данных относятся внутренние и внешние глобальные переменные, константы, пользовательские типы данных, вспомогательные функции и функциональные блоки, без которых работа библиотеки будет невозможна.

3.3.3 Программирование, сборка, тестирование, документирование

В рамках процессов программирования, сборки, тестирования, документирования выполняются следующие работы:

1. Разработка исходного кода программного модуля (модулей) для базовой платформы. За базовую принята платформа Codesys(Astra.IDE).
2. Разработка методики тестирования программного модуля.
3. Добавление разработанного модуля в тестовый проект для базовой платформы. Поскольку тестовый проект привязан к определенной модели ПЛК, разработанный модуль тестируется в проекте с этим ПЛК (например, ПЛК REGUL).
4. Тестирование программного модуля для базовой платформы.
5. Конвертация исходного кода программного модуля для других поддерживаемых платформ. Конвертация требуется для адаптации конструкций языка и смены используемых стандартных библиотек для другой платформы.
6. Добавление модуля в конфигурацию сборки ПО для каждой поддерживаемой платформы.
7. Сборка ПО для каждой поддерживаемой платформы. Результатом сборки является программный пакет (бинарный файл, инсталлятор и т.п.), предназначенный для установки для среды конкретной платформы.
8. Тестирование ПО (и измененных программных модулей в ее составе) для каждой поддерживаемой платформы.
9. Внесение изменений в документацию.
10. Внутренняя приемка ПО, проверка готовности к выпуску релиза: проверка наличия документации, проверка наличия собранных файлов ПО под разные платформы.
12. Выпуск релиза (версии) ПО:
 - выгрузка исходных кодов новых или измененных программных модулей в систему контроля версий;
 - выгрузка тестовых проектов под все платформы в систему контроля версий;
 - выгрузка обновленной документации в систему контроля версий.
 - обновление информации на веб-странице продукта (в том числе выгрузка обновленной документации).

3.4 Эксплуатация

Процесс эксплуатации включает следующие виды работ:

- эксплуатация ПО пользователем (заказчиком);
- техническая поддержка пользователей.

3.4.1 Эксплуатация

Эксплуатацию ПО рекомендуется выполнять в соответствии с документацией пользователя (инструкцией по установке, инструкцией пользователя и другой документацией).

Возможность использования объектов и функций ПО необходимо уточнить в техподдержке, в противном случае правообладатель за неправильное использование ПО ответственности не несет.

Если у заказчика возникают вопросы или проблемы с эксплуатацией, обнаруживаются ошибки, неисправности, то он может запросить техническую поддержку, обратившись к производителю ПО (контактная информация приведена в п. 5).

3.4.2 Техническая поддержка пользователей

При обращении заказчика в техническую поддержку выполняются следующие работы:

1. Регистрация запроса в техподдержку;

2. Изучение и классификация запроса. Все запросы в техподдержку делятся на два типа:

- «консультация»; заказчику требуется помощь в правильном использовании ПО или другая консультация, связанная эксплуатацией ПО;

- «ошибка»; заказчик в процессе эксплуатации обнаружил ошибку и ему требуется ее исправление.

3. Если тип запроса - «консультация», то заказчику формируется ответ по одному из каналов коммуникаций, приведенных в п. 5.2. При необходимости выполняется несколько циклов «запрос-ответ». На этом процесс технической поддержки завершается.

4. Если тип запроса - «ошибка», то сначала выполняется тестирование ПО для выявления и подтверждения наличия ошибки, а затем, если ошибка выявлена, запускается процесс разработки согласно п. 3.3.

Далее заказчику высылается доработанная версия ПО.

5. Закрытие запроса с описанием решения (результата) по этому запросу.

3.4.3 Состав и квалификация персонала

Таблица 1. Персонал для разработки и поддержки библиотеки со стороны правообладателя

Должностные обязанности	Компетенции	Кол-во сотрудников
Разработка библиотеки	Языки программирования МЭК 61831-3 (программирование контроллеров), VB-script, VBA (автоматизация проектирования), системы контроля версий (SVN, GIT). Интегрированные среды разработки: Codesys(Astra.IDE).	1

Таблица 2. Персонал для разработки и поддержки систем на базе библиотеки со стороны заказчика

Вид поставки библиотеки	Должностные обязанности	Компетенции
В составе ПТК или программного комплекса	Эксплуатация ПТК, программного комплекса на базе ПО	Знание эксплуатационной документации на ПТК или программный комплекс
Как отдельный продукт	Разработка программных комплексов или ПТК на базе ПО	Языки программирования МЭК 61831-3 (программирование контроллеров), интегрированная среда разработки под используемую заказчиком платформу (Codesys(Astra.IDE))

4 Решение типовых проблем

В таблице ниже описаны типовые проблемы при эксплуатации ПО и их возможное решение.

№	Проблема	Решение
1	Ошибка компилятора при вызове функционального блока	Не указан один из обязательных входных-выходных («InOut») параметров (например, hmi).
2	Отсутствие реакции блока при подаче команды (не изменяется состояние блока)	Необходимо проверить текущий режим блока (hmi.std.out_mode). Для приема команд блок должен быть в режиме «диспетчер» (для команд высокого приоритета) или «автомат».
3	Блок не изменяет режим при подаче команды смены режима	Проверить привязку и значение входного параметра «plc_control_di». Для восприятия блоком команды смены режима входной параметр должен иметь значение «TRUE».
4	При подаче команды «пуск» блок не переходит в состояние «работа»/«включен»	Проверить привязку параметра «on_di» и приход подтверждающего сигнала на этот вход.
5	Неправильно выполняется расчет выходного параметра «out_x», «out_y» (для блоков «fb_ai», «fb_ao»).	Проверить правильность указания настроечных параметров «hmi.in_sensor», «hmi.in_block_type», «hmi.in_x1», «hmi.in_x2», «hmi.in_y1», «hmi.in_y2».
6	Неверно рассчитываются время наработки блока «hmi.std.out_time_operation» и другие временные параметры.	Проверить, что вызов функционального блока выполняется в циклической задаче (период 100 мсек).

5 Контактная информация производителя программного обеспечения

5.1 Юридическая информация

Информация о юридическом лице компании:

- **Название компании:** ООО «РУ-ДРАЙВ ЦИФРОВЫЕ ТЕХНОЛОГИИ»
- **Юр. адрес:** 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.
- **ОГРН:** 1221600086751
- **ИНН:** 1650418125

5.2 Контактная информация службы технической поддержки

Связаться со специалистами службы технической поддержки можно одним из следующих способов:

- **Сайт:** <https://ru-drive.it>
- **Телефон:** 8 800 555-70-30
- **Email:** digital@ru-drive.com

Фактический адрес размещения инфраструктуры разработки: 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.

Фактический адрес размещения разработчиков: 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.

Фактический адрес размещения службы поддержки: 423800, Республика Татарстан, г. Набережные Челны, Проезд Хлебный, зд. 8/1, производственный корпус №2, каб.201.