СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc154496698)

[1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ 7](#_Toc154496699)

[2. СЛУЖЕБНАЯ ХАРАКТЕРИСТИКА СПЕЦИАЛИСТА 8](#_Toc154496700)

[3. ТЕХНИЧЕСКОЕ ЗАДАНИЕ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 9](#_Toc154496701)

[3.1. Постановка задачи 9](#_Toc154496702)

[3.2. Техническое задание 10](#_Toc154496703)

[3.3. Анализ предметной области 14](#_Toc154496704)

[4. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА 17](#_Toc154496713)

[4.1. Проектирование разрабатываемого программного продукта 17](#_Toc154496714)

[4.1.1. Требования к языкам программирования и среде разработки. Обоснование выбора основных средств разработки и подходов к проектированию 17](#_Toc154496715)

[ЗАКЛЮЧЕНИЕ 18](#_Toc154496716)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc154496717)

# ВВЕДЕНИЕ

Во время моей технологической практики, проведенной в период с 30.10.2023 по 22.12.2023 года в компании ОАО "ВитебскЭнерго", я имел возможность погрузиться в атмосферу рабочего процесса и приобрести ценный опыт в данной области.

Цель проведения данной практики заключалась в формировании профессиональных умений и навыков, а также в расширении и закреплении теоретических знаний, полученных в учебном процессе. Это позволяет студентам лучше понимать производственный труд, связанный с их будущей специальностью.

Основными задачами технологической практики были:

1. **Приобретение профессиональных навыков:** Входе практики я активно участвовал в различных рабочих процессах, что позволило мне освоить основные профессиональные навыки в сфере драйверы теплосчетчиков.
2. **Закрепление и систематизация знаний:** Применение теоретических знаний в практической деятельности способствовало их более глубокому усвоению и пониманию.
3. **Изучение технологии и организации производства:** Я получил возможность познакомиться с современными технологиями и методами организации сбора данных со счетчика, что является важным элементом профессиональной подготовки.
4. **Приобретение практического опыта:** Участие в реальных производственных проектах дало мне ценный опыт, который невозможно получить только из учебников.
5. **Развитие профессионального мышления:** Работа на предприятии требует аналитического и системного подхода, что способствует развитию профессионального мышления.
6. **Повышение уровня квалификации:** Практика позволила мне повысить свой уровень квалификации и лучше подготовиться к будущей профессиональной деятельности.

\*\*Зайкина Т.Г. – руководитель технологической практики от колледжа;

\*Чалко М.П. – руководитель технологической практики от организации.

# ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

ОАО «ВитебскЭнерго» – находиться по адресу: Витебская область, г. Витебск, улица Правды 30.

Структура организации является иерархического типа. Во главе находится директор, ему подчиняются свои собственные подчиненные. Управление ОАО «ВитебскЭнерго» осуществляется на основе сочетания принципов самоуправления коллектива и единоначалия, что означает, каждый сотрудник выполняет приказы и распоряжения руководителя.

Разработка драйвера для теплосчетчика обычно относится к области встраиваемых систем и низкоуровневого программирования. Вот некоторые задачи и ответственности, с которыми может столкнуться разработчик драйвера для теплосчетчика:

1. Взаимодействие с аппаратным обеспечением: Разработчик драйвера должен иметь понимание работы и протоколов взаимодействия с аппаратным обеспечением теплосчетчика. Это может включать работу с различными интерфейсами, такими как UART, SPI или I2C, для чтения и записи данных.

2. Разработка протоколов связи: Теплосчетчики обычно используют специфические протоколы связи для обмена данными с другими устройствами или системами. Разработчик драйвера должен проектировать и реализовывать соответствующие протоколы связи, чтобы обеспечить правильное взаимодействие с теплосчетчиком.

3. Обработка и анализ данных: Драйвер должен обрабатывать данные, полученные от теплосчетчика, и проводить необходимые расчеты для определения параметров, таких как потребление энергии или тепла. Разработчик должен быть знаком с математическими алгоритмами, используемыми для обработки этих данных.

4. Обработка ошибок и исключительных ситуаций: Разработчик должен предусмотреть обработку ошибок и исключительных ситуаций, которые могут возникнуть в процессе работы с теплосчетчиком. Это может включать проверку целостности данных, контрольные суммы, возможность восстановления после сбоев и другие меры для обеспечения надежности работы драйвера.

5. Тестирование и отладка: Разработчик должен проводить тестирование драйвера для проверки его правильной работы и соответствия спецификациям. Это может включать модульное тестирование, интеграционное тестирование и тестирование на реальном оборудовании. Также разработчик должен быть в состоянии отлаживать и исправлять проблемы, которые могут возникнуть в процессе разработки или эксплуатации драйвера.

6. Документирование: Разработчик должен создавать документацию, описывающую функциональность и использование драйвера. Это помогает другим разработчикам понять, как использовать драйвер и интегрировать его в свои проекты.

Обратите внимание, что разработка драйвера для теплосчетчика может быть специфичной для конкретной модели или производителя теплосчетчика. Поэтому важно обратиться к документации и рекомендациям производителя, чтобы уточнить детали и требования для разработки драйвера.

# СЛУЖЕБНАЯ ХАРАКТЕРИСТИКА СПЕЦИАЛИСТА

Для разработки драйвера для теплосчетчика на языке Golang вам потребуется знание следующих технологий и концепций:

1. Golang:

- Основы языка Golang, включая типы данных, переменные, функции, структуры и интерфейсы.

- Работа с пакетами и импортирование зависимостей.

- Обработка ошибок и паник.

- Конкурентное программирование с помощью goroutines и каналов.

2. Протоколы связи:

- Работа с различными протоколами связи, такими как TCP/IP, UDP и HTTP.

- Использование сокетов для установления соединения и передачи данных.

3. Работа с устройствами:

- Взаимодействие с устройствами через последовательный порт или другие интерфейсы.

- Чтение и запись данных с теплосчетчика.

- Обработка протокола обмена данными, используемого теплосчетчиком.

4. Обработка данных:

- Разбор и анализ полученных данных с теплосчетчика.

- Преобразование и обработка данных для отображения и хранения.

5. Тестирование:

- Написание модульных тестов для проверки функциональности драйвера.

- Использование тестовых фреймворков и инструментов для автоматического тестирования.

6. Документация:

- Создание документации и комментариев к коду для облегчения понимания и поддержки драйвера.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Постановка задачи

Постановка задачи для индивидуального задания на тему “Разработка драйвера для теплосчетчика” включает следующие цели и задачи:

Цель: Создание современного и функционального драйвера для дальнейшего использование компанией в производственных целях. Задачи проекта:

Проект разработки драйвера для теплосчетчика включает следующие задачи:

1. Анализ требований и спецификаций:

- Изучение требований к драйверу, предоставленных производителем теплосчетчика.

- Анализ особенностей и протокола связи теплосчетчика.

2. Проектирование архитектуры драйвера:

- Определение структуры и функций драйвера, необходимых для взаимодействия с теплосчетчиком.

- Разработка алгоритмов обмена данными с теплосчетчиком.

- Выбор программного или аппаратного подхода к реализации драйвера.

3. Реализация драйвера:

- Написание кода драйвера, соответствующего спецификациям и требованиям.

- Интеграция драйвера с операционной системой или программной платформой, на которой он будет работать.

- Тестирование функциональности и корректности работы драйвера.

4. Оптимизация и оптимизация производительности:

- Оптимизация работы драйвера для улучшения его производительности и эффективности.

- Проведение тестирования и анализа производительности драйвера.

5. Тестирование и отладка:

- Проведение тестирования драйвера для проверки его функциональности и соответствия требованиям.

- Отладка и исправление возможных ошибок и проблем в работе драйвера.

6. Документирование:

- Создание документации, описывающей функциональность, использование и настройку драйвера.

- Подготовка инструкций по установке и настройке драйвера для конечных пользователей.

7. Защита информации:

- Обеспечение безопасности данных и защиты от несанкционированного доступа к драйверу и связанным с ним данным.

8. Поддержка и обновление:

- Предоставление поддержки пользователю, отвечая на вопросы и решая проблемы, связанные с драйвером.

- Возможность выпуска обновлений и исправлений для драйвера, если необходимо.

9. Соблюдение стандартов и нормативных требований:

- Убедиться, что драйвер соответствует применимым стандартам и нормативным требованиям.

10. Тестирование и демонстрация:

- Проведение тестового использования драйвера с реальным теплосчетчиком, чтобы проверить его работу в реальных условиях.

- Демонстрация и предоставление драйвера заказчику для оценки и проверки его соответствия требованиям и ожиданиям.

## Техническое задание

Краткое описание для разработки драйвера теплосчетчика:

1. Анализ требований:

- Изучение документации и спецификаций теплосчетчика для определения функциональных возможностей и протоколов связи.

- Понимание основных параметров, которые должны быть доступны через драйвер.

2. Разработка архитектуры драйвера:

- Определение структуры и интерфейса драйвера для обмена данными с теплосчетчиком.

- Разработка алгоритмов для чтения и записи данных из/в теплосчетчик.

3. Реализация драйвера:

- Написание кода драйвера в соответствии с определенной архитектурой.

- Разработка обработчиков для обмена данными по протоколу связи теплосчетчика.

- Интеграция драйвера с операционной системой или программной платформой.

4. Тестирование и отладка:

- Проведение тестирования драйвера для проверки его функциональности и корректности работы.

- Отладка и исправление возможных ошибок и проблем в работе драйвера.

5. Документирование:

- Создание документации, описывающей функциональность, использование и настройку драйвера.

- Подготовка инструкций по установке и настройке драйвера для конечных пользователей.

6. Защита информации:

- Обеспечение безопасности данных и защиты от несанкционированного доступа к драйверу и связанным с ним данным.

7. Поддержка и обновление:

- Предоставление поддержки пользователю, отвечая на вопросы и решая проблемы, связанные с драйвером.

- Возможность выпуска обновлений и исправлений для драйвера, если необходимо.

8. Соблюдение стандартов и нормативных требований:

- Убедиться, что драйвер соответствует применимым стандартам и нормативным требованиям.

9. Тестирование и демонстрация:

- Проведение тестового использования драйвера с реальным теплосчетчиком, чтобы проверить его работу в реальных условиях.

– Демонстрация и предоставление драйвера заказчику для оценки и проверки его соответствия требованиям и ожиданиям.

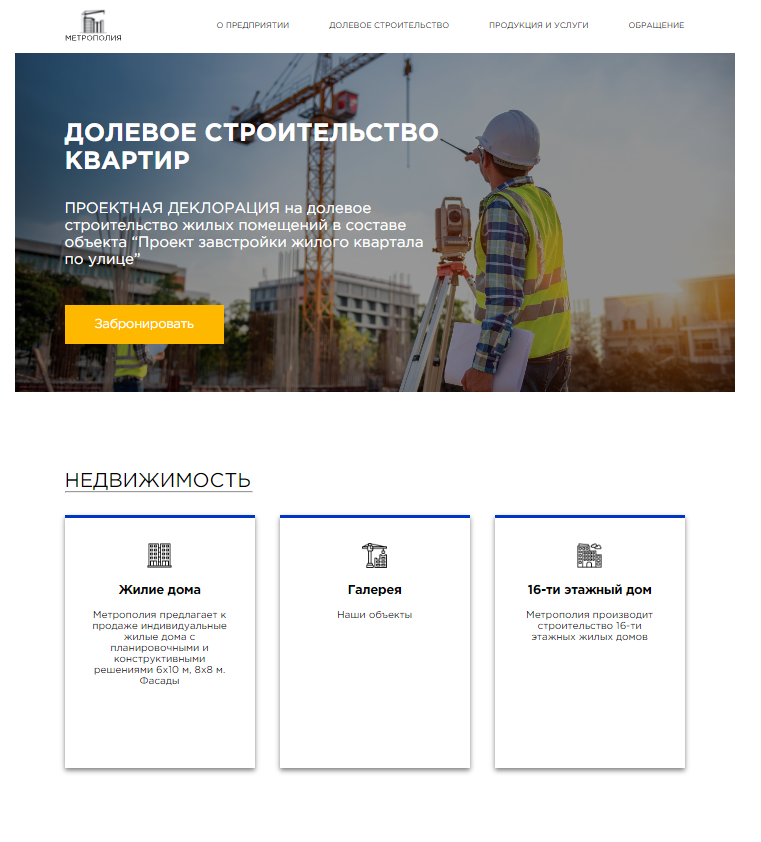


Рисунок 1 – Пример разрабатываемого драйвера

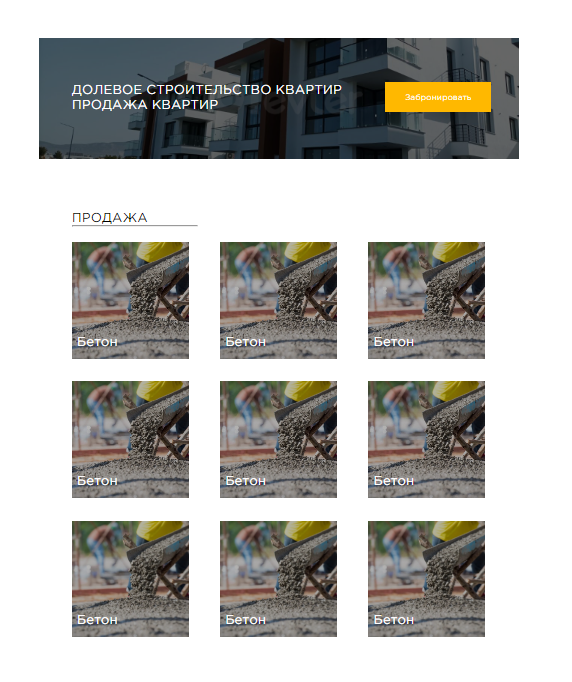


Рисунок 2 – Пример разрабатываемого драйвера

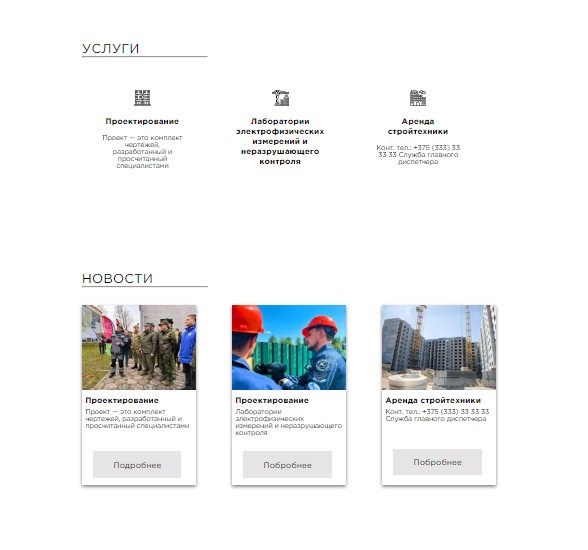


Рисунок 3 – Пример разрабатываемого драйвера

## Анализ предметной области

Для разработки драйвера для теплосчетчика на языке Golang, вам следует руководствоваться следующими шагами:

1. Изучение спецификации теплосчетчика:

- Получите доступ к документации, предоставленной производителем теплосчетчика.

- Понимание протокола обмена данными, используемого теплосчетчиком, и формата данных, которые необходимо считывать и записывать.

2. Настройка окружения разработки:

- Установите Go на свою систему и настройте рабочую среду.

- Разберитесь с инструментами и библиотеками, которые могут быть полезны при разработке драйвера для теплосчетчика.

3. Создание основного модуля драйвера:

- Создайте новый проект Golang и определите основной модуль драйвера.

- Импортируйте необходимые библиотеки и зависимости для работы с протоколом обмена данными и устройством.

4. Установка соединения с теплосчетчиком:

- Используйте соответствующие библиотеки для установки соединения с теплосчетчиком, используя доступные интерфейсы (например, последовательный порт, TCP/IP).

5. Чтение и запись данных:

- Реализуйте функции для чтения и записи данных с теплосчетчика, в соответствии с протоколом обмена данными.

- Обработайте ошибки и исключительные ситуации при взаимодействии с устройством.

6. Обработка данных:

- Разберитесь с форматом данных, получаемых от теплосчетчика, и преобразуйте их в удобный формат для дальнейшей обработки или отображения.

- Реализуйте логику для расчета показаний и других операций, связанных с теплосчетчиком.

7. Тестирование и отладка:

- Напишите модульные тесты для проверки функциональности драйвера.

- Отладьте и исправьте ошибки, которые могут возникнуть в процессе разработки.

8. Документация и комментарии к коду:

- Добавьте комментарии к коду, объясняющие его функциональность и логику.

- Создайте документацию, описывающую использование и API вашего драйвера.

9. Интеграция и поддержка:

- Интегрируйте ваш драйвер с другими компонентами системы, в которой он будет использоваться.

- Предоставьте техническую поддержку и обновления драйвера по мере необходимости.

# ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

## Проектирование разрабатываемого программного продукта

### Требования к языкам программирования и среде разработки. Обоснование выбора основных средств разработки и подходов к проектированию

1. Язык программирования Go (Golang): Выбор языка программирования Go обусловлен его простотой, эффективностью и мощными возможностями. Вот почему Go является привлекательным выбором:

- Простота использования: Go разработан с упором на простоту и понятность синтаксиса. Это позволяет разработчикам быстро освоить язык и создавать эффективный код.

- Высокая производительность: Go компилируется в машинный код, что обеспечивает высокую производительность выполнения программ. Он имеет встроенную поддержку параллельных вычислений, благодаря чему может эффективно использовать многоядерные системы.

- Конкурентность: Go имеет встроенную поддержку горутин (goroutines) и каналов (channels), что делает легким создание конкурентных программ. Это позволяет разрабатывать эффективные и отзывчивые программы, которые могут обрабатывать множество задач одновременно.

- Большая стандартная библиотека: Go поставляется с обширной стандартной библиотекой, которая включает в себя множество полезных пакетов для работы с сетью, файлами, шифрованием, тестированием и многим другим. Это значительно упрощает разработку приложений, так как многие необходимые функции уже встроены в язык.

- Кросс-платформенность: Go поддерживает множество операционных систем и архитектур, включая Windows, macOS и Linux. Это позволяет создавать кросс-платформенные приложения, которые работают на разных платформах без необходимости внесения больших изменений в код.

- Богатая экосистема: Go имеет активное сообщество разработчиков, которое создает и поддерживает множество сторонних библиотек и инструментов. Это позволяет расширить функциональность языка и упростить разработку приложений.

В целом, язык программирования Go (Golang) является простым, эффективным и мощным языком, который позволяет разработчикам создавать производительные и конкурентные приложения. Он имеет большую стандартную библиотеку, активное сообщество и пользуется популярностью в различных областях разработки, включая веб-серверы, микросервисы, системное программирование и многое другое.

2) Среда разработки JetBrains GoLand - это интегрированная среда разработки (IDE) от компании JetBrains, специально разработанная для языка программирования Go (Golang). Вот почему она отлично подходит для разработки на Golang:

1. Полная поддержка Golang: GoLand обеспечивает полную поддержку языка Go, включая подсветку синтаксиса, автодополнение, переходы по определениям, быстрое исправление ошибок, форматирование кода, анализ кода и интеграцию с инструментами Go. Это значит, что IDE предоставляет все необходимые функции для удобной и эффективной разработки на Golang.

2. Инструменты и функции разработки: GoLand предлагает богатый набор инструментов и функций разработки, таких как отладчик, система контроля версий, автоматическое завершение кода, рефакторинг, генерация кода, интеграция с пакетным менеджером Go и многое другое. Это помогает ускорить разработку и повысить производительность.

3. Интеграция с другими инструментами JetBrains: GoLand полностью интегрирована с другими продуктами JetBrains, такими как IntelliJ IDEA, что позволяет разработчикам работать с проектами на нескольких языках программирования в единой среде разработки. Это особенно удобно, если у вас есть проекты, требующие разработки на разных языках.

4. Настройка и расширение: GoLand предлагает множество настроек, которые позволяют адаптировать среду разработки под ваши потребности. Кроме того, у вас есть возможность расширить функциональность IDE с помощью плагинов, созданных сообществом разработчиков JetBrains.

В целом, среда разработки GoLand от JetBrains является мощным инструментом для разработки на Golang, предоставляющим все необходимые функции и инструменты для эффективной разработки приложений на этом языке.орменные десктопные приложения, используя веб-технологии.

# ЗАКЛЮЧЕНИЕ

За время прохождения технологической практики в компании «ВитебскЭнерго» я улучшил свои теоретические и практические навыки, углубил свои знания в языке программирования Golang, библиотеке Git, научился разрабатывать драйверы и преодолевать трудности в разработке. Познакомился с командной работой, организацией, структурой и принципами функционирования предприятия.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация по языку программирования Golang [Электронный ресурс]. - https://go.dev/doc/
2. Документация Альфа-Мера для теплосчетчика [Электронный ресурс]. - https://alphamera.by/dokumentacija-i-po/
3. Документация по библиотеке Git [Электронный ресурс]. -   
   https://git-scm.com/doc
4. Маккарти С., Хантер Дж. Unity in Action, Third Edition - Manning Publications, 2018. — 624 с.
5. Мартин Р. Чистый код: Создание, анализ и рефакторинг - Питер, 2010. - 464 с.