

Министерство науки и высшего образования Российской Федерации
Лысьвенский филиал федерального государственного автономного образовательного
учреждения высшего образования
Пермский национальный исследовательский политехнический университет
Факультет профессионального образования

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему «Разработка проекта автоматизированного дозатора жидкостей с
микроконтроллерным управлением»
студента группы КСК-18-1спо по специальности
09.02.01 Компьютерные системы и комплексы
Максимова Константина Васильевича _____

Руководитель работы: _____ С.А. Зыкин
Консультант по
экономической части: _____ К.В. Кондратьева
Консультант по промышленной экологии
и охране труда: _____ А.К. Тороцин
Рецензент: _____ (_____)
Допуск к защите: _____ М.Н. Апталаев

Лысьва, 2022 г

Министерство науки и высшего образования и Российской Федерации
Лысьвенский филиал федерального государственного автономного образовательного
учреждения высшего образования
«Пермский национальный исследовательский политехнический университет»
ПЦК «Естественнонаучных дисциплин»

Утверждаю:

Председатель ПЦК
_____ М.Н. Апталаев
«__» _____ 2022 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

студента Максимова Константина Васильевича курса 4
группы КСК9-18-1спо
специальность 09.02.01 Компьютерные системы и комплексы

Тема задания «Разработка проекта автоматизированного дозатора жидкостей с микроконтроллерным управлением»

Структура выпускной квалификационной работы такова:

а) Введение. Аргументировать актуальность выбранной темы, ее теоретическое значение и практическую значимость, сформулировать цель и конкретные задачи исследований. Конкретизировать объект и предмет исследований.

б) Исследовательский раздел. Понятие автоматического дозатора напитков. Критерии при выборе дозатора напитков и жидкостей. Основные способы дозирования жидкостей. Обзор существующих на рынке решений. Анализ требований к разрабатываемому устройству.

в) Конструкторский раздел. Структурная схема разрабатываемого устройства. Выбор комплектующих для разрабатываемого устройства. Разработка программы и устройства.

г) Охрана труда и промышленная экология. Анализ вредных и опасных факторов на рабочем месте инженера-электронщика. Разработка средств защиты от воздействия выбранного ВиОПФ. Экологические требования к утилизации вычислительной и оргтехники, а также их расходных материалов.

д) Организационно-экономический раздел. Выполнение технико-экономической оценки разработанного устройства.

е) Заключение. Краткое изложение решенных задач, актуальность работы, соответствие полученных результатов теме и заданию ВКР.

ж) Список использованных источников.

з) Приложения.

Дата выдачи _____

Срок окончания _____

Руководитель ВКР

_____ /С.А. Зыкин/

«__» _____ 2022 г.

Задание утверждено на заседании ПЦК «Естественнонаучных дисциплин» протокол № _____ 2022 г.

Председатель ПЦК _____ /М.Н. Апталаев/

«__» _____ 2022 г.

Министерство науки и высшего образования Российской Федерации
Лысьвенский филиал федерального государственного автономного образовательного
учреждения высшего образования
«Пермский национальный исследовательский политехнический университет»

ПЦК «Естественнонаучных дисциплин»

Утверждаю
Председатель ПЦК
М.Н. Апталаев
«15» февраля 2022 г.

График выполнения выпускной квалификационной работы

студента группы КСК9-18-1сно

специальности 09.02.01 Компьютерные системы и комплексы

Максимова Константина Васильевича

(фамилия, имя, отчество)

Содержание	Сроки	Отметка выполнении	Дата							Подпись
			08.03-20.04	21.04-30.04	01.04-17.05	18.05-25.05	26.05-30.05	31.05-06.06	07.06-15.06	
Введение	08.03 –									
1. Теоретическая часть	20.04									
2. Охрана труда и БЖД	20.04 –									
3. Промышленная экология	30.04									
4. Инженерно-технической часть	01.05 –									
5. Организационно-экономическая часть	17.05									
Оформление дипломного проекта: Заключение Список используемой литературы Приложения	18.05- 25.05									
Выполнение чертежей, оформление демонстрационных материалов к ВКР	26.05- 30.05									
Представление проекта на проверку и отзыв руководителя Нормоконтроль	31.05 –									
Подготовка доклада и демонстрационных материалов к предзащите	06.06									
Предоставление работы на проверку председателю ПЦК	04.06 –									
Предзащита ВКР Иметь к преддипломной защите: - рецензия нормоконтроля; - отзыв руководителя; - подпись руководителя по экономической части	06.06 – 07.06									

<ul style="list-style-type: none"> - подпись руководителя по охране труда - подпись руководителя по промышленной экологии - презентация - доклад 																			
Устранение замечаний по всей ВКР	08.06-15.06																		
Рецензирование Сдача работ на кафедру	20.06																		
Диск с материалами ВКР	21.06																		
Защита ВКР	22.06 – 23.06																		

Руководитель ВКР _____ / _____ /

« ____ » _____ 2022 г.

Студент _____ / _____ /

« ____ » _____ 2022 г.

Согласовано:

Председатель ПЦК «Естественнонаучных дисциплин» _____ / М.Н. Апталаев /
« ____ » _____ 2022 г.

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

IP – Классификация способа защиты, обеспечиваемого оболочкой технического устройства от доступа к опасным частям, попадания внешних твёрдых предметов и воды и проверяемого стандартными методами испытаний

LCD – экран на основе жидких кристаллов

UART – универсальный асинхронный приемопередатчик

КПД – коэффициент полезного действия

САПР – система автоматизированного проектирования

Энкодер – датчик угла поворота

РЕФЕРАТ

Максимова К.В. Разработка проекта автоматизированного дозатора жидкостей с микроконтроллерным управлением, выпускная квалификационная работа: стр. 57, рис. 10, табл. 10, 20 источников, 29 формул.

ДОЗИРОВАНИЕ, ЖИДКОСТЬ, НАПИТОК, ЁМКОСТЬ, ПОДАЧА, АВТОМАТИЗАЦИЯ, РАЗРАБОТКА, СИСТЕМА, АВТОМАТИЗИРОВАННОЕ УСТРОЙСТВО, МИКРОКОНТРОЛЛЕР, ИНТЕРФЕЙС, СТРУКТУРНАЯ СХЕМА, ПРИНЦИПИАЛЬНАЯ СХЕМА.

Объектом исследования ВКР является способы розлива жидкостей.

Цель работы – Разработка проекта автоматизированного дозатора жидкостей с микроконтроллерным управлением.

В процессе проектирования были получены следующие результаты:

- Создана структурная и принципиальная схема автоматизированного устройства дозирования жидкостей;
- Создан макет, по средствам имеющихся компонентов и материалов, который способен подключаться к компьютеру;
- Изучены программные интерфейсы для написания кода;
- Изучены программы для построения чертежей и принципиальной схемы;
- Получены навыки процесса сборки, правильного подключения.

В результате работы был разработан проект автоматизированного дозатора жидкостей с микроконтроллерным управлением.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ.....	4
1.1 Понятие автоматического дозатора напитков.....	4
1.2 Критерии при выборе дозатора напитков и жидкостей:	4
1.3 Основные способы дозирования жидкостей.....	6
1.4 Обзор существующих на рынке решений	14
1.5 Анализ требований к разрабатываемому устройству	18
2 КОНСТРУКТОРСКИЙ РАЗДЕЛ	20
2.1 Структурная схема разрабатываемого устройства.....	20
2.2 Выбор комплектующих для разрабатываемого устройства	21
2.3 Разработка принципиальной схемы и корпуса устройства	25
2.4 Разработка управляющей программы для микроконтроллера.....	29
3 ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ	32
3.1 Техника безопасности на рабочем месте инженера-электроника.....	32
3.2 Опасные производственные факторы на рабочем месте инженера-электроника	36
3.3 Методы утилизации вычислительной техники.....	38
3.4 Вывод по разделу	39
4 ОРГАНИЗАЦИОННО–ЭКОНОМИЧЕСКАЯ ЧАСТЬ.....	41
4.1 Расчет затрат на разработку программы для микроконтроллера.....	41
4.2 Расчет затрат на внедрение программы для микроконтроллера.....	44
4.3 Расчет эксплуатационных текущих затрат по программе для микроконтроллера	46
4.4 Расчет экономической целесообразности разработки и внедрения информационных технологий.....	48
ЗАКЛЮЧЕНИЕ.....	54
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	55
Приложение А – Программный код	57

ВВЕДЕНИЕ

Автоматический дозатор напитков – это устройство, которое разливает негазированные безалкогольные напитки по стаканчикам разного объема 220 мл, 330 мл, 500 мл. В современном мире такие электронные дозаторы обладают большой популярностью и это неудивительно, т. к. дозатор может выступать интересным и удобным решением, как для домашнего использования, так и для праздничных столов. Дозаторы изготавливают как для бытового использования, так и для продажи. Преимуществом является простота восполнения емкости запаса жидкости, интуитивно понятная эксплуатация, удобная компоновка. Использование автоматического дозатора позволяет свести физические усилия человека к минимуму, что позволяет человеку ленивому, больному или с ограниченными возможностями не совершать лишних телодвижений и получать необходимое количество напитка, также дозатор может выступать необычной вещью в интерьере дома, корпус имеет гибкий дизайн и может подойти к любому интерьеру.

Объектом исследования ВКР является способы розлива жидкостей.

Предметом исследования ВКР является автоматизация процесса розлива жидкостей.

Целью выпускной квалификационной работы является разработка проекта автоматизированного дозатора жидкостей с микроконтроллерным управлением.

Для достижения цели были выделены следующие задачи:

- рассмотреть и изучить аналогичные устройства дозирования;
- выполнить анализ требований к устройству;
- подобрать подходящую элементную базу, разработать структурную и принципиальную схему устройства, написать алгоритм работы и программную часть для микроконтроллера;
- выполнить технико-экономическое обоснование проекта.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Понятие автоматического дозатора напитков

Зачастую люди во время смешивания напитков сталкивались с проблемой неаккуратного использования емкостей для напитков, а именно с проливанием жидкости. Также не у всех имеется ёмкость для дозирования жидкости в текущий момент времени или она вовсе отсутствует. С разработкой автоматического дозатора можно не только забыть о всех ранее перечисленных проблемах, но и создавать разнообразные коктейли, что удивит друзей и близких, а также подарит им массу эмоций.

Автоматический дозатор работает с самыми распространенными объёмами, такими как 220 мл, 330 мл и 500мл, что позволит выбирать желаемый объём стакана. С помощью ручки на дозаторе можно выбрать нужное соотношение двух напитков, после чего остается только поставить стакан на платформу дозатора и нажать кнопку разлива. Система автоматически определит объём стакана, а далее выдаст точные пропорции жидкости. Максимальная вместимость системы — это один стакан, что идеально подходит для индивидуального использования, но не подходит для компании из нескольких лиц, т.к. расход жидкости будет большой и необходимо будет постоянно пополнять резервуары. Данный дозатор позволяет полностью избежать потерь жидкости. В качестве питания система использует повербанк, благодаря чему дозатор является мобильным и его легко можно использовать в различных местах пребывания человека, например на улице, или дома.

1.2 Критерии при выборе дозатора напитков и жидкостей:

- вязкость разливаемой жидкости;
- используемый тип ёмкости для разлива;
- время наполнения;
- объём ёмкости;
- технологии разлива жидкости.

1.2.1 Вязкость разливаемой жидкости

Как подсолнечное масло, так и спирт разливается в ёмкость при помощи дозатора. Подсолнечное масло и яблочный сок являются жидкостями, но в тоже время имеют абсолютно разную вязкость. Устройства для заполнения ёмкостей подсолнечным маслом в корни отличается, от устройств, работающих с яблочным соком.

Поршневой дозатор хорошо справляется с процессом разлива подсолнечного масла, поскольку он сохраняет температуру жидкости, Жидкость, обладающая низкой вязкостью, осуществляет движения посредством движения пневматического цилиндра. В то время как яблочный сок можно разливать самотеком в переливной дозатор.

При выборе решения для разлива жидкостей для начала нужно понимать характеристики вашего продукта, чтобы устройство удовлетворяло ваши потребности. Необходимо определить, какую вязкость имеет ваш продукт, будь то обычная вода или паста. Отталкиваясь от характеристик вязкости, это поможет сделать выбор устройства разлива. Такой параметр, как вязкость затрагивает несколько областей, а именно тип системы дозирования, скорость наполнения и другие смежные области в использовании дозаторов. Определение вязкости продукта – достаточно важный этап в выборе систем дозирования.

1.2.1 Используемый тип ёмкости для разлива

Представление вида ёмкости, в которую следует наполнить жидкость, является немаловажным фактором при выборе наилучшего устройства для разлива. Нужно понимать какую ёмкость вы заполняете. Сделанную из стекла или же пластика, квадратную или круглую, низкую или высокую, с небольшим горлышком либо же с крупным, устойчивую или неустойчивую – здесь представлена лишь малая часть характеристик, затрагивающих вид ёмкости. Данные характеристики оказывают влияние не только на разработку, но и на конструкцию систем дозирования.

1.2.2 Время наполнения

Логично, что одним из более значительных вопросов со стороны производства: «Сколько ёмкостей можно заполнить за период работы?». Что представляет собой наилучшее решение: одиночный ручной полуавтомат, промежуточное автоматическое решение для полной линии или автоматизированная система с использованием роторов? Время заполнения находится в зависимости от таких факторов как: вязкость жидкости, вид ёмкости, объем заполнения, а также немаловажным факторов является способность жидкости к образованию пены.

1.2.3 Объём ёмкости

Понимать необходимый объем наполнения звучит довольно обычно. Однако же это достаточно немаловажно при определении различных связанных характеристик, а именно скорость заполнения, вязкость жидкости и способность к образованию пены у жидкости. В зависимости от отрасли вас могут возникнуть определенные требования отраслевых стандартов. Одним из таких требования является высокая точность. Таким образом, точное понимание необходимых объемов заполнения и требований в отрасли обладают решающим значением для производственной эффективности.

1.2.4 Технологии разлива жидкости

Для выбора технологии разлива продукта нужно знать вид продукта. Определенные жидкости обладают разными характерными чертами, к примеру они имеют восприимчивость к сдвигу или же помимо жидкости имеют в своём составе различные твердые частицы. От данных характеристик зависит выбор дозаторов, к примеру дозаторы с разными насосами, с синхронизированным потоком подачи, дозаторы с возможностью вакуумирования, поршневые и перистальтические дозаторы, которые выполняют подачу продукта посредством выдавливания.

1.3 Основные способы дозирования жидкостей

В ходе развития процесса дозирования жидкостей были изобретены различные методы дозирования, которые имеют разную точность, а также

подходящих для различных видов жидкостей и консистенций. Все способы дозирования сталкиваются с проблемой общего свойства: жидкости никак не текут под одинаковым давлением в трубе, т.к. плотность жидкости меняется в зависимости от количества примесей, газирования и температурных показателей. Из этого следует, что точное количество жидкости невозможно просчитать на основании сечения трубы и времени разлива.

Существуют 4 способа дозирования жидкости, данные способы имеют разные параметры, а именно цена, скорость, точность разлива, а также гигиеничность. Для разных жидкостей используются разные методы дозирования. Дорогостоящие косметические и парфюмерные изделия требуют высокоточного дозирования, питьевые напитки требуют соблюдения всех гигиенических требований, каждая жидкость также отличается разными физико-химическими свойствами, а именно плотностью, вязкостью, поверхностным натяжением, температурой замерзания и наличием примесей в составе. К примеру, сок, подсолнечное масло и молоко, они имеют разные физико-химические характеристики и состав. Сравним физико-химические свойства представленных напитков в таблице 1.

Таблица 1 – Физико-химические свойства жидкостей

	Яблочный сок	Подсолнечное масло	Молоко
Плотность	1033 – 1053 кг/м ³	900 – 980 кг/м ³	1026-1032 кг/м ³
Вязкость	~ 1,5 мПа · с	при 20°С = 58 – 92 мПа·с	при 20°С = 1,6 – 2,1 мПа·с
Поверхностное натяжение	~ 0,0725 Н/м	при 20 °С = 0,03 – 0,033 Н/м	при 20 °С = 0,0424 – 0,051 Н/м

Продолжение таблицы 1

Температура застывания	~ -1,67 °С	от -30 до -1°С	от -0,53 до 0,55 °С
Возможные примеси	отсутствуют	отсутствуют	примеси комбикорма, пылевых частиц и сгустков белка молока

На основании таблицы мы можем убедиться, что каждая из представленных жидкостей и вправду имеет разные свойства, что в свою очередь влияет на метод дозирования. К примеру, если нужно дозировать молоко, в котором имеются примеси и последует соблюдать ряд определенных условий для этого. А если нужно будет дозировать апельсиновый сок, содержащий мякоть, то отсюда выдвигается ряд других условий, например выбор подходящей помпы для разлива.

Рассмотрим некоторые методы дозирования жидкостей:

- дозирование по весу с использованием тензодатчика
- дозирование с использованием датчика уровня
- дозирование с измерением объема;
- дозирование при помощи перистальтического насоса;
- дозирование при помощи датчика массового расхода.

1.3.1 Дозирование по весу с использованием тензодатчика.

Представленный метод называется гравиметрическим, в этом методе применяется тензометрический датчик, принцип его работы заключается в измерении массы ёмкости и её силы воздействия, а также жидкости. При полном наполнении ёмкости тензодатчик отслеживает поданное количество жидкости в ёмкость в режиме реального времени. По итогу достижения определенного количества жидкости, контроллер веса останавливает процесс её подачи. Тензодатчики могут быть представленным как в цифровом, так и в аналоговом виде. В датчике, который имеет цифровой вид находится встроенный преобразователь сигнала, в то время как сигнал с аналогового тензодатчика проходит процесс оцифровки непосредственно в весовом терминале, в связи с чем возникают задержки.

Процедура заполнения ёмкости, представлена данным образом:

Сначала ёмкость помещается на платформу для измерения веса, далее выполняется взвешивание ёмкости без жидкости, после чего происходит наполнение жидкостью до определенного веса.

Конвейерные линии производят процесс заполнения одновременно из нескольких сопл. В среднем на заполнение одной литровой ёмкости уходит примерно от 5 до 10 секунд, при соблюдении безошибочного определения точности веса.

Основным преимуществом дозирования при помощи тензодатчика можно выделить тот факт, что автоматическое измерение веса каждой отдельно взятой ёмкости перед наполнением позволит определять поврежденные и несоответствующие своему весу ёмкости и удалять их с конвейерной ленты.

Недостатком данного метода является время заполнения жидкости в сравнении со средней статистикой выполнения данной операции. Возрастание веса во время наполнения может сигнализировать об переполнении заданного объёма и вследствие об утечке из ёмкости, благодаря чему можно сделать вывод, что необходима полная остановка конвейерной ленты для устранения её загрязнений.

Отсюда следует, что использование данного метода дозирования позволяет вести контроль дополнительных параметров работы, в связи с чем увеличивается автоматизация контроля разлива.

Дозирование по весу позволяет выполнять разлив высокой точности, с точностью до грамма. Процесс калибровки тензодатчика проходит благодаря гилям эталонного веса, что и обеспечивает высокую точность дозирования. Данные систем, измеряющих вес в цифровом виде, упрощают программирование контроллеров автоматизации производства, установку подходящих параметров скорости заполнения жидкости, подготовку отчётов о сделанной работе, контролирование производственных графиков.

Дозирование по весу благополучно используется как в пищевой, так и в химической промышленности, поскольку использование в системе тензодатчика исключает контакт с жидкостью дозирования. Тензодатчики для данного метода дозирования выпускаются с довольно высокой степенью защиты, а конкретно до IP68. Защита IP68 – говорит о том, что датчик может работать в довольно тяжелых условиях, он обладает защитой от случайного попадания пыли и обладает возможностью долгосрочного погружения под определенным давлением. Наличие данной защиты у датчика позволит выполнять процедуру промывания ёмкостей под давлением. Принцип работы этого метода дозирования представлен на рисунке 1.

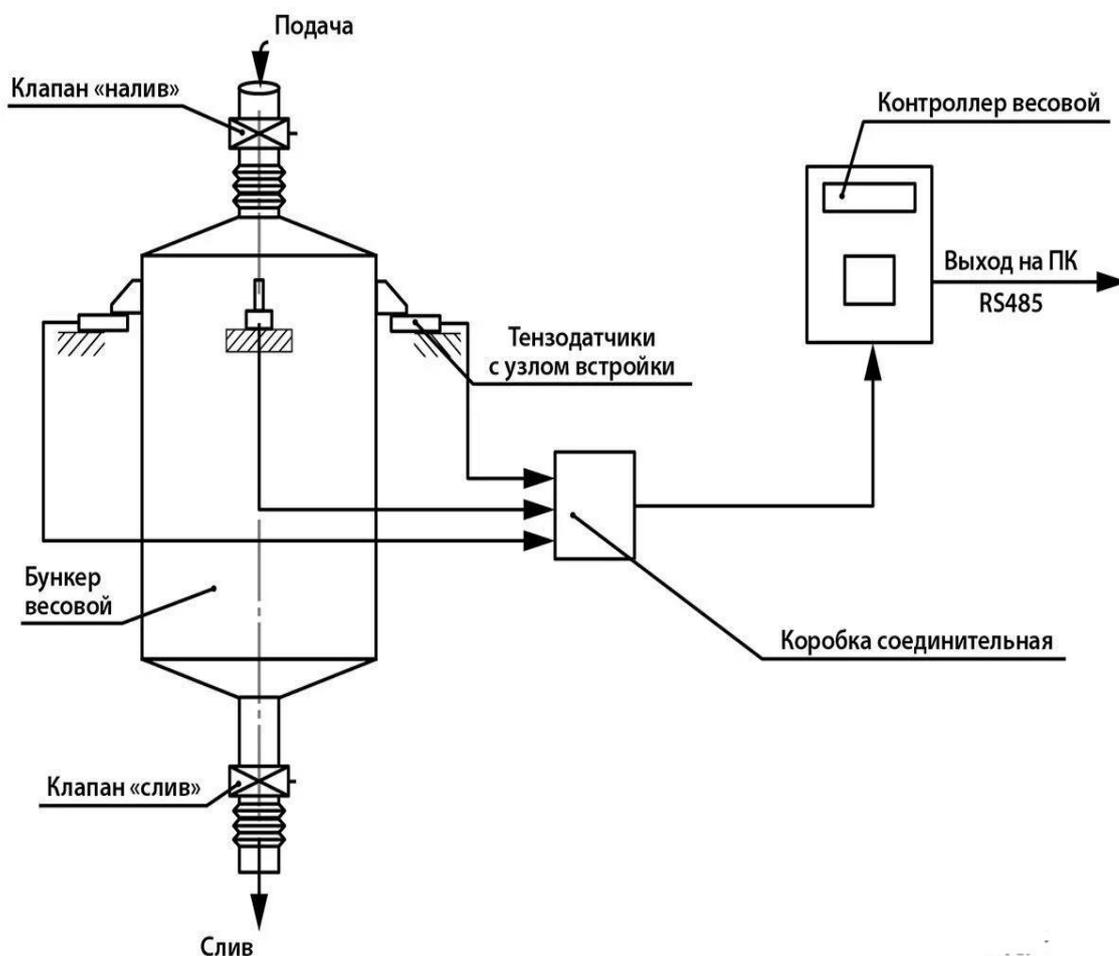


Рисунок 1 – Принцип дозирования по весу с использованием тензодатчика

1.3.2 Дозирование с использованием датчика уровня

Данный способ контролируется достижением жидкости верхней заданной точкой в ёмкости, для осуществления данной задачи датчик уровня устанавливается в горлышко ёмкости и заполнения жидкостью происходит до её контакта с датчиком уровня, иным способом работы датчика является реагирование на изменение объёма жидкости со внешней стенки ёмкости.

В случае нахождения датчика в горлышке ёмкости становится возможным дозирование токопроводящих жидкостей, а именно жидкость может содержать в составе соли.

Различные минеральные и растительные масла не содержат определенное количество соли, отсюда следует, что их дозировка этим методом невозможна.

Представленный способ дозирования имеет более низкую точность, поскольку ёмкости имеют разный внутренний объём, к примеру, стеклянные ёмкости зачастую имеют такой объём. Этот метод дозирования в основном используют для фасования никак не дорогостоящих товаров. Контактное датчика с подаваемой жидкостью полностью исключит возможность гигиенического разлива, поскольку датчик перемещается из одной ёмкости в другую.

Этот метод отлично подходит для изобарического разлива. Изобарический розлив – используется для фасовки жидкостей, которые в своём составе содержат углекислый газ и являются газированными, например различные лимонады. Данный метод отлично подходит для газированных напитков, поскольку принцип его работы заключается в том, что давление в трубке дозатора и ёмкости имеют равное значение.

Рассмотрим принцип работы устройства изобарического разлива с вакуумированием ёмкости на рисунке 2.

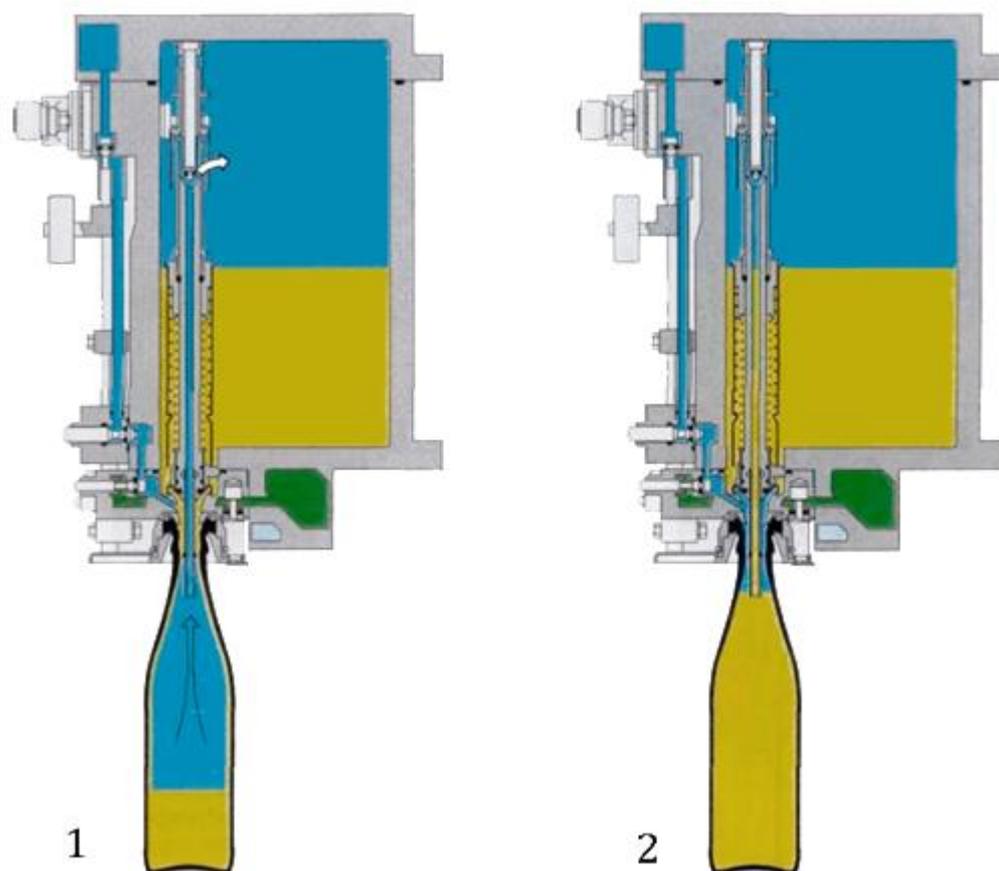


Рисунок 2 – Принцип работы устройства изобарическим методом с вакуумированием ёмкости. На первом этапе подаётся жидкость и поднимается к соплу, на втором этапе она касается сопла и процесс останавливается

На рисунке можно увидеть, каким образом происходит заполнение ёмкости.

Большим преимуществом этого метода является визуально одинаковый уровень жидкости в бутылке, что может хорошо сказаться для продаж в магазинах.

1.3.4 Дозирование с измерением объема

Метод заключается в измерение объема протекающей через сопло жидкости магнитно-индуктивным способом.

Под воздействием магнитного поля выполняется расщепление ионов жидкости, которая проходит через сопло, ионы, проходящие на электродах датчика, создают разность потенциалов. Созданное напряжение позволяет рассчитать объемную скорость потока. Данный способ является актуальным только для токопроводящих жидкостей. Под каждый вид дозируемой

жидкости приходится выполнять персональное калибрование, поскольку в составе разных жидкостей содержится различное количество ионов.

1.3.5 Дозирование при помощи перистальтического насоса

Главной особенностью такого метода дозирования является использование перистальтического дозатора. Благодаря чему отсутствует контакт между дозируемым продуктом и элементами дозатора. Продукт располагается внутри силиконовой трубки и под воздействием, на которую вращающиеся ролики пережимают головки насоса и происходит выдавливание продукта по трубке из резервуара в ёмкость. Зная изначальный объём внутреннего сечения трубки и угол поворота роликов в головке, контроллер дозатора высчитывает нужное количество вращений роликов, для подачи заданного объёма продукта. Отталкиваясь от отличительных особенностей этого метода, можно сделать вывод, что его в основном используют для розлива стерильных, агрессивных и сильно пачкающих жидкостей, смесей и паст. Принцип работы такого метода дозирования представлен на рисунке 3.

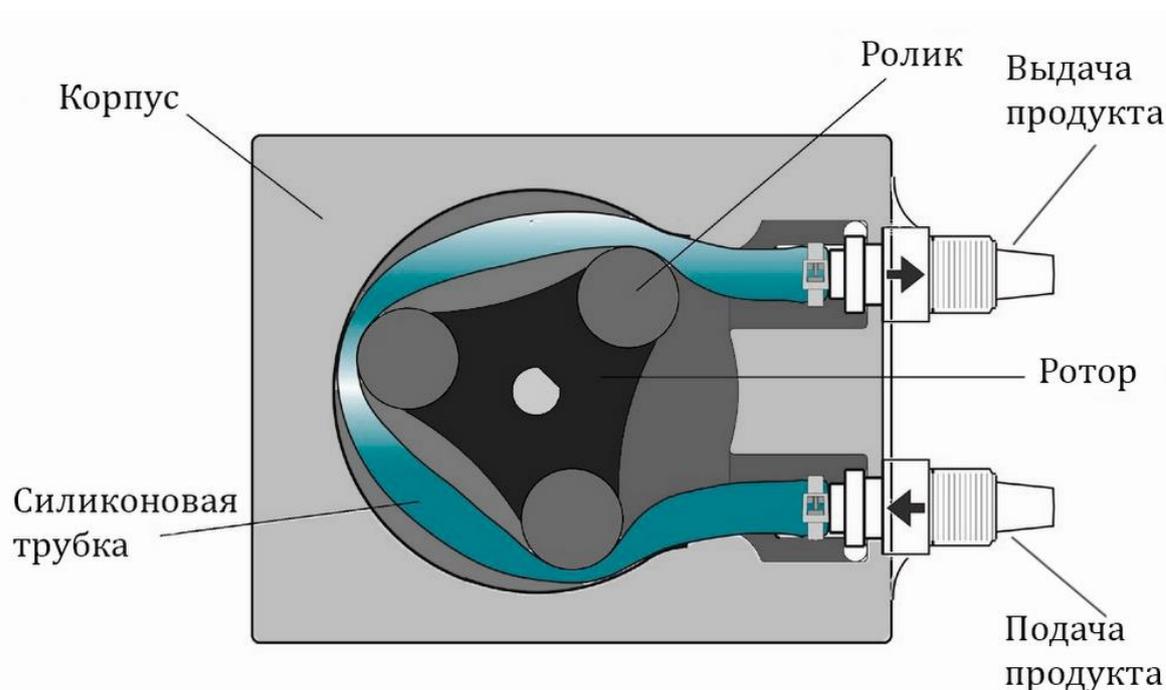


Рисунок 3–Принцип работы роликового перистальтического насоса

1.3.6 Дозирование при помощи датчика массового расхода

Данный метод дозирования использует в своём основании принцип силы Кориолиса. Он заключается в том, что жидкость течет по двум вибрирующим трубам, воздействие силы Кориолиса вызывает фазовый сдвиг в этих вибрациях. На основании полученных данных система рассчитывает массу продукта. Этот метод может использоваться для подачи не токопроводящих продуктов.

Данный метод дозирования является дорогостоящим, поскольку процесс калибрования датчика при его производстве является очень трудоемким.

1.4 Обзор существующих на рынке решений

BaroBot – это роботизированное устройство, которое предназначается для подачи напитков. Такого рода устройство помогает создавать коктейли, подавать безалкогольные напитки и содовые. Данное устройства вмещает в себя 12 ёмкостей. Представленная разработка подаст и перемешает нужные жидкости и выдаст как сложный коктейль, так и обычный напиток.

Корпус устройства выполнен из акрилового стекла, благодаря чему он не подвергается воздействию жидкостей и тяжелых ёмкостей, а также бактерий и микроорганизмов. Питание устройства осуществляется по бытовой сети, что в свою очередь ограничивает его мобильность. Данное устройство имеет светодиодную подсветку из-за чего его можно использовать в плохом освещении. Управление осуществляется как вручную на корпусе, так и удаленно в специальном приложении при помощи телефона. Программное обеспечение устройства имеет интуитивно понятный интерфейс, а в базе данных находится от тысячи рецептов различных напитков. Основными особенностями устройства являются вариативность в выборе ёмкостей, возможность взбалтывания жидкостей, возможность использования в качестве жидкости сиропов, газированных напитков и различных соков.

Электронная часть устройства выполнена из общедоступных деталей, что позволяет самостоятельно осуществлять замену комплектующих.

Вид устройства можно рассмотреть на рисунке 4.



Рисунок 4 – Barobot

Ключевым недостатком данного устройства является использование каретки для перемещения ёмкостей, а также цена, поскольку она составляет 199 000 рублей.

Домашний бар Drinkworks – это устройства для приготовления классических безалкогольных напитков с использованием специальных одноразовых капсул. Желаемая жидкость заливается в резервуар, после чего в специальный отсек вставляется одноразовая капсула с каким-либо сиропом и выполняется смешивание коктейля. Данное устройства вмещает в себя всего одну ёмкость для заполнения.

Корпус Drinkworks визуально неотличим от машины для приготовления кофе. Ширина лицевой части и боковых стенок составляет 34,29 см, высота устройства составляет 33,02 см, Корпус имеет плоское квадратное основание и прямоугольным вид. Корпус покрашен темно-серым

цветом и имеет хромированную стойку на правой стороне лицевой части, которая переходит в нависающий отсек для капсул и выходное сопло. На левой стороне передней панели располагается прозрачный резервуар для жидкости, который можно снять для очистки и наполнения, а также на передней панели располагается синий LCD экран с сенсорным управлением. Данный прибор хорошо подходит как для домашнего использования, так и для различных офисов.

Внешний вид устройства представлен на рисунке 5.

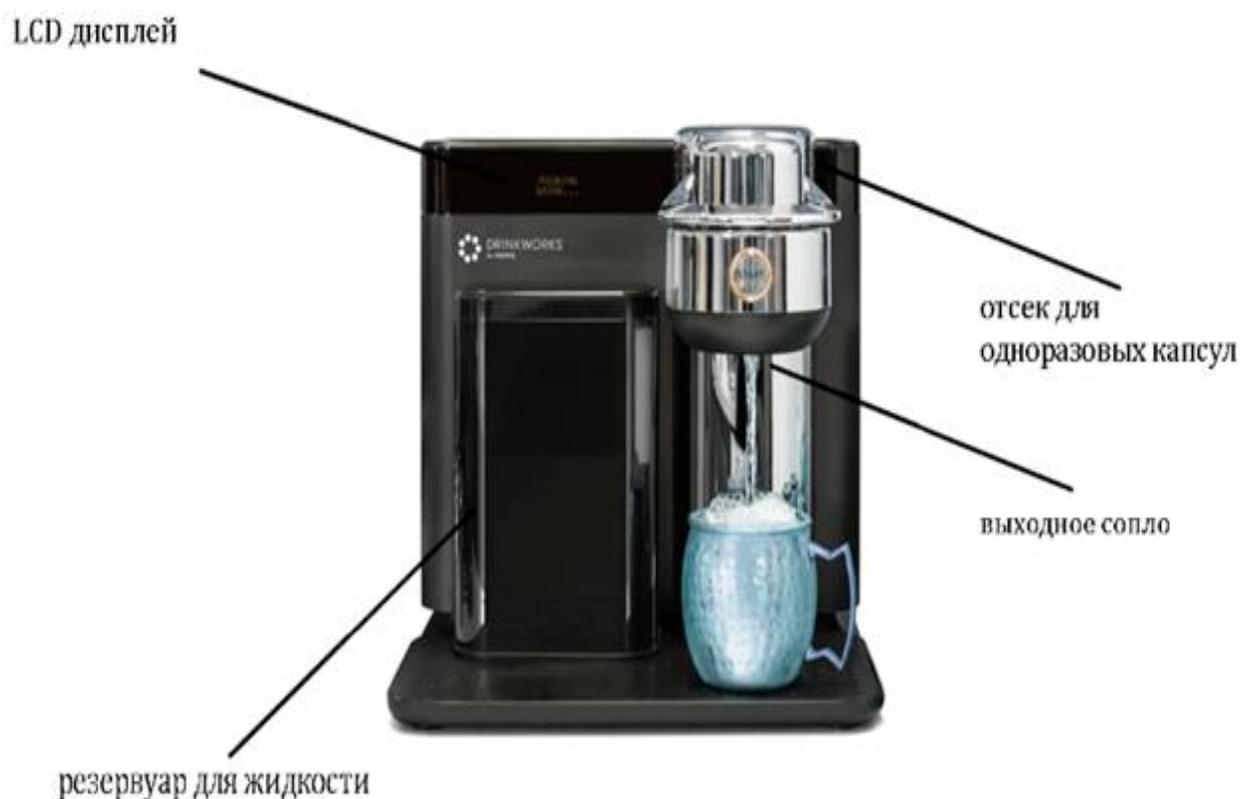


Рисунок 5 – Внешний вид домашнего бара Drinkworks

Основным недостатком данного устройства является использования специальных одноразовых капсул, что довольно ограничивает возможности его эксплуатации, т.к. капсулы с сиропом являются расходным материалом и их постоянно придется докупать. Цена за упаковку из 4 капсул составляет 1 857 рублей. Также из недостатков можно выделить малый объём резервуара, он составляет всего 1 литр, что в свою очередь заставляет его пополнять при частном использовании. Цена данного устройства составляет 34 824 рубля.

Робот бармен для разлива напитков

Робот бармен для разлива напитков – представляет устройство для автоматической подачи жидкости по заданному количеству мили литров. Вместимость данного устройства составляет 6 ёмкостей объёмом 220 мл. Заполнения происходит выполняется только в ёмкости, установленные на специальное место, что исключает возможно пролития жидкости. Устройство питается от встроенного аккумулятора формата 18650, что позволяет ему работать автономно от сети.

Основными преимуществами данного устройства является наличие уникальных функций, а именно наличие звукового сопровождения в виде различных тостов и радио. Данный робот бармен имеет возможность необычной подачи жидкости так называемой игрой в русскую рулетку, когда напиток наливается в любую случайную ёмкость и в американскую, когда жидкость подаётся случайно в каждую ёмкость, кроме одной. Также данный робот бармен имеет возможно подачи жидкости сразу в несколько ёмкостей. Робот имеет светодиодную подсветку в местах для установки ёмкости.

Длина устройства составляет 32 см, ширина 22 см и высота 26 см. Корпус полностью выполнен из фанеры, обработанной яхтным полуматовым лаком. На платформе для установки ёмкостей имеются специальных места для их установки. Переход между ёмкостями осуществляется при помощи сервопривода, который предварительно калибруется под места их установки. После установки ёмкости на место для разлива, нужно выбрать желаемый объём жидкости и сервопривод направит сопло к таре и начнёт процесс подачи напитка с помощью помпы.

Данный прибор хорошо подходит для домашнего использования, а также может является отличным функциональным украшением для различных праздничных мероприятий.

Внешний вид устройства представлен на рисунке 6.

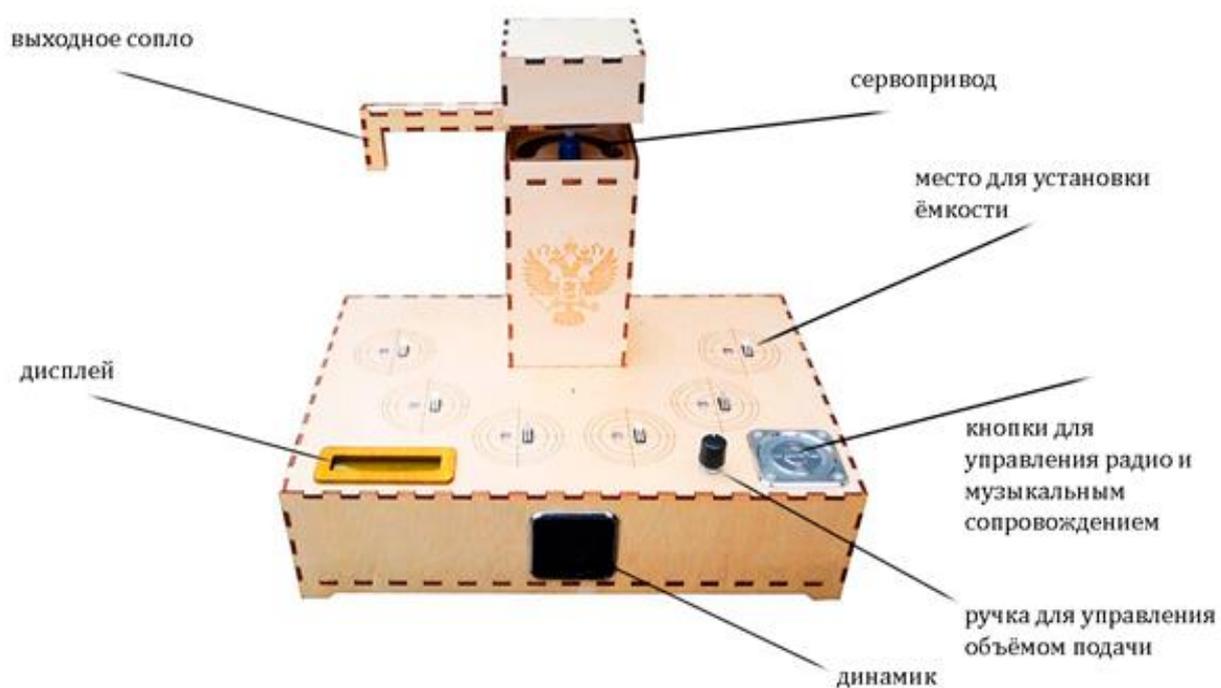


Рисунок 6 – Внешний вид робота-бармена для разлива напитков

Основными недостатками данного устройства является отсутствие возможности создания коктейлей путём смешивания жидкостей и использование ёмкостей объёма свыше 220 мл.

Цена данного устройства составляет 12 000 рублей.

1.5 Анализ требований к разрабатываемому устройству

Проектируемое устройство будет актуально для любителей точного дозирования и смешивания жидкостей для создания необычных коктейлей. Устройство предназначается для индивидуального использования одним лицом. Разрабатываемое устройство подходит людям ленивым или же с ограниченными возможностями. Назначением разработки устройства дозирования и смешивания заключается в выставке.

Устройство дозирования должно осуществлять точное дозирование, при помощи помп, а также осуществлять смешивание двух жидкостей. Управление интерфейсом разрабатываемого устройства должно осуществляться при помощи датчика угла поворота и кнопки. В устройстве должна присутствовать возможность выбора пропорций смешивания жидкостей.

Устройство должно питаться от мощного повербанка с выходным напряжением 3,7 – 5 В и током 1,5 – 2 А. Для увеличения времени работы нужно использовать повербанк с большой ёмкостью.

На корпусе устройства необходимо расположить порт для подключения повербанка, датчик угла поворота, инфракрасный датчик для определения наличия ёмкости, кнопку и LCD экран.

В корпуса разрабатываемого устройства должны быть расположены функциональные блоки питания, исполнительные устройства управления и подачи жидкости в сопло.

Требования к эксплуатации и хранению разрабатываемого устройства, заключаются в поддержании необходимой температуры, лежащей в диапазоне от -5 до +30 градусов, в случае хранения температура может достигать +40 градусов. Эксплуатация и хранение устройства в помещении с относительно влажностью воздуха больше 80% не рекомендуется, значения влажности воздуха ниже 30% так же является нежелательным. Жидкость для дозирования должна иметь следующие характеристики: низкая вязкость и отсутствие твёрдых частиц. Может использоваться слабо газированная жидкость. Подходящими жидкостями для смешивания и дозирования могут выступать гранатовый и яблочный сок. Выполнение данных условий позволит избежать появления неисправностей в помпах.

Поверхность для расположения устройства должна быть ровной и устойчивой, падение не предусмотрено.

Основными преимуществами проектируемого устройства являются простота эксплуатации, функция смешивания жидкостей, мобильность и ценовая доступность.

2 КОНСТРУКТОРСКИЙ РАЗДЕЛ

2.1 Структурная схема разрабатываемого устройства

Для начала разработки устройства необходимо выделить в системе структурные блоки, их выполняемые функции и взаимосвязь между собой. После этого будет создана структурная схема устройства.

Для разрабатываемого устройства выделяются следующие блоки: блок питания, управления, блок подачи жидкости, и блок обнаружения ёмкости.

Блок питания включает в себя аккумуляторную батарею (повербанк) с подключёнными к ней разъёмом питания.

В блок управления входит модуль энкодера, кнопка, LCD дисплей 1602 и Arduino Nano. Интерфейс реализуется на LCD дисплее. Данный блок позволяет взаимодействовать с интерфейсом устройства и задавать желаемое количество жидкости для подачи. Arduino Nano отвечает за управление основными функциями системы, а также она предназначена для сбора и обработки информации.

В блок подачи жидкости входит драйвер и две помпы, задачей драйвера является усиление входного напряжения и управление помпами.

Отдельным блоком можно выделить блок обнаружения ёмкости. Данный блок имеет в своём составе инфракрасный датчик для сбора данных о нахождении ёмкости на платформе для установки.

Взаимосвязь между блоками может быть, как однонаправленной, так и двунаправленной. Разница между этими понятиями заключается в направлении передачи сигнала или данных. В случае однонаправленного сигнала данные всегда передаются от одного блока к другому, и поддержка обратного ответа не предусмотрена, в случае двунаправленной связи, все блоки, соединённые этой связью, могут быть как приёмниками, так и отправителями.

Связь между блоком управления и блоком подачи жидкости является однонаправленной, и имеет направление от блока взаимодействия с

интерфейсом, к блоку подачи жидкости. Обратная связь в этом случае не имеет смысла.

Связи с блоком питания установлены со всеми остальными блоками, поскольку каждому элементу необходимо питание.

Блок обнаружения ёмкости имеет однонаправленную связь с блоком управления. На основе данных, полученных в этом блоке, микроконтроллер будет выполнять последующий алгоритм подачи жидкости.

На рисунке 7 изображена полученная структурная схема разрабатываемого устройства.

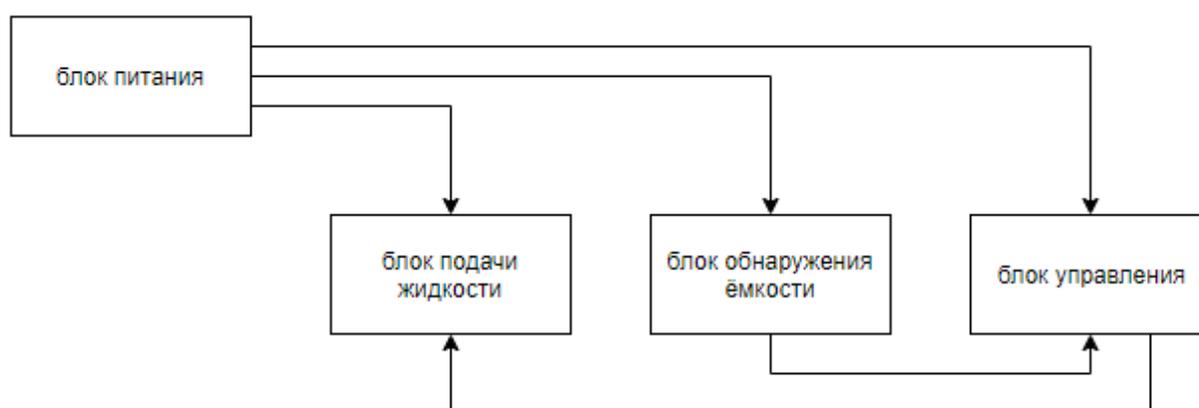


Рисунок 7 – Структурная схема устройства

На рисунке 7 можно видеть направление потока данных и сигналов, обозначенное стрелочкой. После того, как сигнал от блока обнаружения ёмкости передаётся на блок управления, пользователь при помощи интерфейса выполняет калибровку и устанавливает желаемое для подачи количество жидкости и далее сигнал от микроконтроллера передаётся на блок подачи жидкости.

Все остальные связи между блоками показанные на рисунке 7 представляют собой передачу питания системы.

2.2 Выбор комплектующих для разрабатываемого устройства

Следующим шагом необходимо выбрать модели используемых в проекте элементов и комплектующих.

Для начала будет выбран микроконтроллер. Из сравниваемых микроконтроллеров будут выбраны варианты с максимальным показателем объёма флеш-памяти, значение которой должно быть минимум 16 Кб, и тактовой частоты. К тому же микроконтроллер должен иметь минимум 15 линий ввода-вывода, и обладать встроенным интерфейсом UART. Немаловажным фактором при выборе будет цена.

К сравнению представлены микроконтроллеры семейства ATmega, а именно ATmega328p, ATmega2560, ATmega32u4, ATmega168P, ATmega644. Для сравнения были выбраны эти микроконтроллеры по причине того, что они удовлетворяют требованиям по минимальному количеству линий ввода-вывода и интерфейсу UART. Далее данные микроконтроллеры будут сравниваться по своей производительности и цене. Сравнение перечисленных микроконтроллеров представлено в таблице 2.

Таблица 2 – Сравнение микроконтроллеров

Название	Atmega328p	Atmega2560	Atmega32u4	ATmega168	ATmega644
Цена, р	150	600	300	250	500
Тактовая частота, МГц	16	16	16	20	20
Флеш-память, Кб	32	256	32	16	64
Линии ввода-вывода	20	54	20	20	32
UART	1	4	1	1	1

После выполнения сравнения параметров микроконтроллеров был выбран микроконтроллер ATmega328p, поскольку его стоимость гораздо ниже аналогов, и он обладает хорошим быстродействием. У данного микроконтроллера подходящее количество цифровых и аналоговых выходов и он не обладает избыточной производительностью и идеально подходит для устройства по критерию цена/производительность.

Следующим будет выбран модуль драйвер для управления помпами.

Задачами устройства является подача жидкости, при помощи помп, отсюда следует, что требования, предъявляемые к модулю драйвера, будут связаны с параметрами напряжения и тока.

Для модуля драйвера выдвигаются требования по напряжению для моторов, напряжению для управляющих контактов, рабочий и максимальный ток для каждого мотора. Количество управляющих контактов должно быть равно двум или более, и питание модуля должно составлять не более 7 вольт. Для сравнения между собой были выбраны следующие модули: MX1508, XY-160D, L298N.

Рассмотрим модуль MX1508. Его максимальное напряжение для моторов составляет 10 вольт, а минимальное 2 вольта. Напряжение для управляющих контактов составляет 1,8-7 вольт и максимальные ток для каждого мотора составляет 2,5 ампера, количество управляющих контактов 2 и имеется возможность подключения двух моторов, данные характеристики подходят под выдвинутые требования к модулю.

Модуль XY-160D. Максимальный ток для мотора составляет 7А. Также поддерживается двухканальный режим работы. Напряжение для моторов составляет 6,5–27 вольт. А напряжение для управляющих контактов составляет 3-6,5 вольт. Данный модуль не соответствует требованиям, т.к. обладает избыточной мощностью.

Последний рассматриваемый модуль L298N имеет рабочее напряжение 5 вольт. Напряжение на выход 5-35 вольт, максимальные ток 2 ампера. Данный модуль обладает радиатором для пассивного охлаждения. Минимальное питание модуля 5 вольт не подходит по установленным требованиям.

Сравнение этих модулей также представлено в таблице 3.

Таблица 2 – Сравнение модулей драйвера

	MX1508	XY-160D	L298N
Цена	54 р.	364 р.	66 р.
Напряжение для моторов	2–10 вольт	6,5–27 вольт	5–35 вольт
Напряжение для управляющих контактов	1,8–7 вольт	3–6,5 вольт	5 вольт
Рабочий ток для каждого мотора	1,5 ампера	2,5 ампера	1,5 ампера
Максимальный ток для каждого мотора	2,5 ампера	7 ампер	2 ампера
Количество подключаемых моторов	2	2	2

По результатам сравнения модулей драйвера был выбран модуль MX1508 по причине подходящих входных и выходных значений напряжения и силы тока, а также в данный модуль строена защита от останковки моторов, что в свою очередь увеличивает надежность, как самого драйвера, так и подключенных к нему двигателей. Микросхема обладает встроенной тепловой защитой с автоматическим восстановлением по мере снижения температуры до рабочих величин. Драйвер меньше нагревается, за счет встроенных в мостовую схему MOSFET транзисторов с низким сопротивлением перехода, отсюда достигается высокий КПД, по сравнению с другими MX1508 самой низкой ценой и остальные модули не удовлетворяют требованиям напряжения и силы тока входных и выходных характеристик, пусть и имеют некоторые положительные факторы в других параметрах.

Так же необходимо выбрать помпу для подачи жидкости. Помпа должна иметь питание 12 вольт, иметь мембранный насос, помпа должна состоять полностью из пищевых материалов и работать без применения масла. Помп с представленными характеристиками довольно большое количество, и они отличаются лишь качеством исполнения и ценой, поэтому мы выбираем помпу соответственно критерию цена/качество.

Далее нужно выбрать дисплей для устройства. В качестве дисплея для устройства идеально подходит жидкокристаллический дисплей 1602. Данный дисплей может отображать 2 линии на 16 символов, имеет питание 5 вольт и

размер модуля составляет: 80 мм x 35 мм x 11 мм. Данный дисплей полностью выполняет все требования к интерфейсу устройства.

Финальный список комплектующих для разрабатываемого устройства:

- Плата Arduino nano V3;
- Энкодер WAVGAT 5V;
- Переключатель pbs33b 12V;
- Дисплей LCD 1602;
- Модуль гнездового преобразователя MICRO USB/DIP;
- Драйвер MX1508;
- Инфракрасный датчик;
- 2 мембранных пищевых помпы для жидкости DC12V.

2.3 Разработка принципиальной схемы и корпуса устройства

Для начала нужно выбрать САПР для разработки принципиальной схемы устройства. В качестве рассматриваемых программ будут взяты EasyEDA, DipTrace и Kicad. К данным САПР выдвигаются следующие требования: возможность разработки печатной платы по принципиальной схеме, автотрассировка, возможность добавления элементов пользователем, отображение печатной платы в формате 3D и возможность экспорта полученных схем в формат png.

EasyEDA – САПР для разработки принципиальной схемы, с возможностью перевода схемы в режим разработки печатной платы. Имеется возможность автотрассировки дорожек платы, готовый вариант можно посмотреть в 3D варианте плате. Имеется online версия программы, для которой требуется лишь авторизация, без необходимости скачивать приложение.

DipTrace – программный продукт позволяющий выполнять разработку принципиальных схем и печатных плат. Предоставляет возможность просмотра разработанной печатной платы в формате 3D модели, с

установленными элементами. При работе имеется возможность использовать автотрассировщик.

Kicad – это САПР для разработки как принципиальной схемы, так и печатной платы различных устройств. В программе имеется возможность работы с различными существующими элементами схемы, а также добавления своих. САПР не поддерживает возможности автотрассировки, а также не может выводить полученную печатную плату в 3D форме.

Сравнение представленных программ рассматриваем в таблице 4.

Таблица 4 – Сравнение программ для разработки принципиальной схемы

	EasyEDA	DipTrace	Kicad
Возможность добавления пользователями элементов схемы	+	+	+
Разработка печатной платы	+	+	+
Отображение 3D модели разработанной печатной платы	+	+	-
Возможность экспорта принципиальной схемы и схемы печатной платы в png	+	+	+
Возможность автотрассировки	+	+	-

Проанализировав таблицу 4, можно сделать вывод что наиболее подходящим вариантом для разработки принципиальной схемы и печатной платы является САПР EasyEDA, поскольку Kicad не удовлетворяет требованиям возможности автотрассировки и отображения печатной платы в 3D форме, а DipTrace не имеет возможности работы в online режиме, это потребует времени на установку.

После выбора программы для разработки принципиальной схемы, начинается разработка принципиальной схемы, на которой обозначаются все используемые комплектующих, их контакты и обозначения, также на принципиальной схеме выстраивается связь между контактами элементов схемы.

При проектировании принципиальной схемы необходимо выполнить несколько основных подсистем, к которым относится система питания, система подачи жидкости, система управления, и система определения ёмкости.

На рисунке 8 изображена разработанная принципиальная схема устройства со всеми подключенными системами.

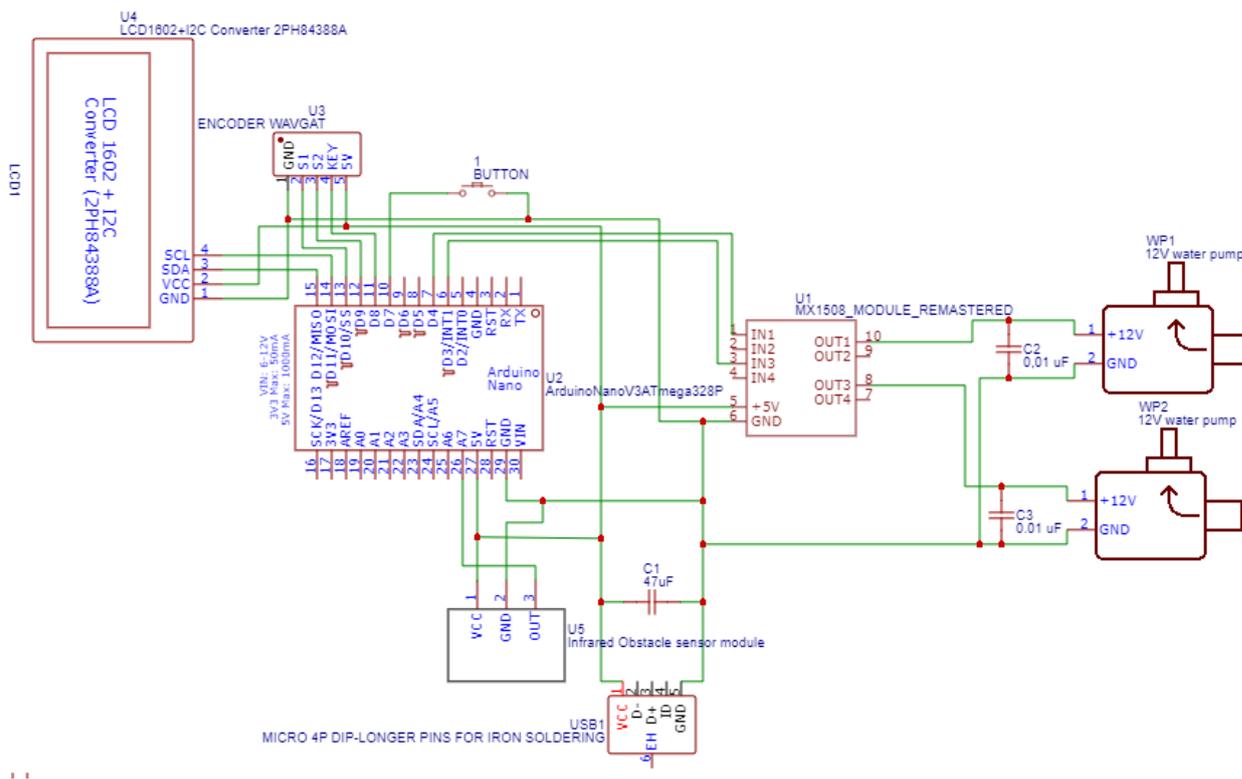


Рисунок 8 – Принципиальная схема устройства

Основным источником питания системы является внешний аккумулятор (повербанк), который имеет номинальное напряжение ~ 5 вольт и силу тока равную $\sim 1,5$ ампера. Источник питания подключается к модулю гнездового преобразователя MICRO USB/DIP, с помощью USB/MICRO USB кабеля. На схеме модуль гнездового преобразователя обозначается как USB1. На выходы преобразователя подключается конденсатор ёмкостью $47\mu\text{F}$ (C1) для стабилизации и сглаживания напряжения.

К системе подачи жидкости относится драйвер MX1508 для управления двумя помпами, который обозначает как U1, а также в систему входят две помпы, которые обозначаются WP1 и WP2. После получения сигнала для подачи от микроконтроллера, драйвер подаёт нужно напряжение на помпы для подачи жидкости.

К системе определения ёмкости относится инфракрасный датчик, которой обозначается как U5. Его функция довольно проста, когда ёмкость находится на площадке для установки датчик передает сигнал на микроконтроллер.

В систему управления входит lcd экран, на схеме подписан как U4, энкодер (U3), кнопка (1), а также микроконтроллер (U2), который обрабатывает информацию и управляет всеми системами. На lcd экране реализуется интерфейс для взаимодействия с устройством, в котором с помощью энкодера можно выбрать желаемое количество первой и второй жидкости для подачи, после выбора желаемого объёма жидкости нужно подтвердить выбор с помощью кнопки, но предварительно нужно выполнить калибровку системы путем зажатия кнопки на энкодере. Прогнав воду через сопло, приступаем к основному этапу калибровки, а именно замеряем за сколько секунд подается определенный объём жидкости и вписываем получившиеся время в программный код устройства. Калибровка выполняется единовременно, и после дальнейшая калибровка устройства не требуется.

В качестве корпуса для проектируемого устройства подойдет пластиковый ящик из хозяйственного магазина размером 230x160x75 мм. Ящик должен иметь петли, которые способны удерживать крышку под углом 90°.

Также необходимо второе дно для закрытия электронных компонентов и для создания посадочной площадки под ёмкость. В качестве второго дна подойдет прямоугольная деревянная фанера или же прямоугольная доска из пластика. Размер второго дна определяется по внутреннему размеру ящика, а толщина второго должна быть не более 2 см. Примерный чертеж второго дна представлен на рисунке 9.

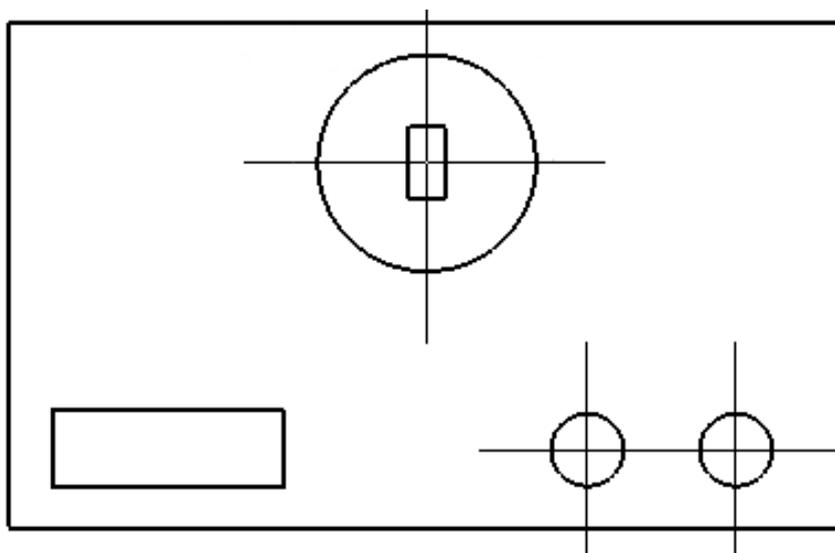


Рисунок 9 – Чертеж второго дна

На чертеже расположены отверстия для установки кнопки, энкодера, дисплея, концевого выключателя и 4 отверстия для трубок подачи жидкости. Энкодер и кнопку прикручиваем специальными гайками для крепления в отверстия, остальные комплектующие фиксируем термоклеем. В соответствии с принципиальной схемой припаиваем провода к комплектующим.

2.4 Разработка управляющей программы для микроконтроллера

Для разработки управляющей программы необходимо выбрать среду разработки программного кода для микроконтроллера. Для этого будет выполнен выбор среди сред: Micro chip, MPLAB, WinAVR.

К рассматриваемым средам разработки выдвигаются данные требования: встроенный компилятор для C++, встроенный программатор, возможность отладки, возможность симулирования работы микроконтроллера, работа на Windows 10 и способ распространения.

Micro chip – среда разработки от компании AVR, которая раньше носила название Atmel Studio. Среда разработки позволяет писать код на языке программирования C++ или на Assembler. Имеется встроенный симулятор работы микроконтроллера, который может показать, что записано в регистры и память микроконтроллера, с учётом разных моделей.

MPLAB – среда разработки являющаяся набором программ для работы с кодом микроконтроллера. MPLAB позволяет писать и отлаживать код на C++ или Assembler, для этого имеется встроенный компилятор кода.

WinAVR – среда разработки для написания кода микроконтроллеров различных семейств. В данном программном продукте имеется возможность написания и отладки кода, без возможности отображения состояния регистров. Интерфейс программы наиболее прост в освоении и работе.

Сравнение рассмотренных сред разработки показано в таблице 5.

Таблица 5 – Сравнение функций сред разработки

	Micro chip	MPLAB	WinAVR
Встроенный компилятор для C++	+	+	+
Возможность отладки	+	+	+
Возможность симуляции	+	+	-
Свободно распространяемое ПО	+	+	+
Встроенный программатор	+	+	+
Поддержка работы на Windows 10	+	+	-

Выполнив анализ таблицы 5, была выбрана среда разработки micro chip, по причине того, что она наиболее подходит под требуемые задачи. MPLAB является более серьёзным и сложным в освоении инструментом. WinAVR не имеет симуляции, а также имеет проблемы при работе на Windows 10.

После выбора среды разработки был выбран язык программирования Arduino Wairing, который имеет схожих синтаксис с C++, имея лишь дополнительные функции, позволяющие сделать написание кода наиболее удобным. Для написания программного кода необходимо разработать блок-схему алгоритма работы устройства. На рисунке 10 можно видеть блок-схему алгоритма работу программы для микроконтроллера.

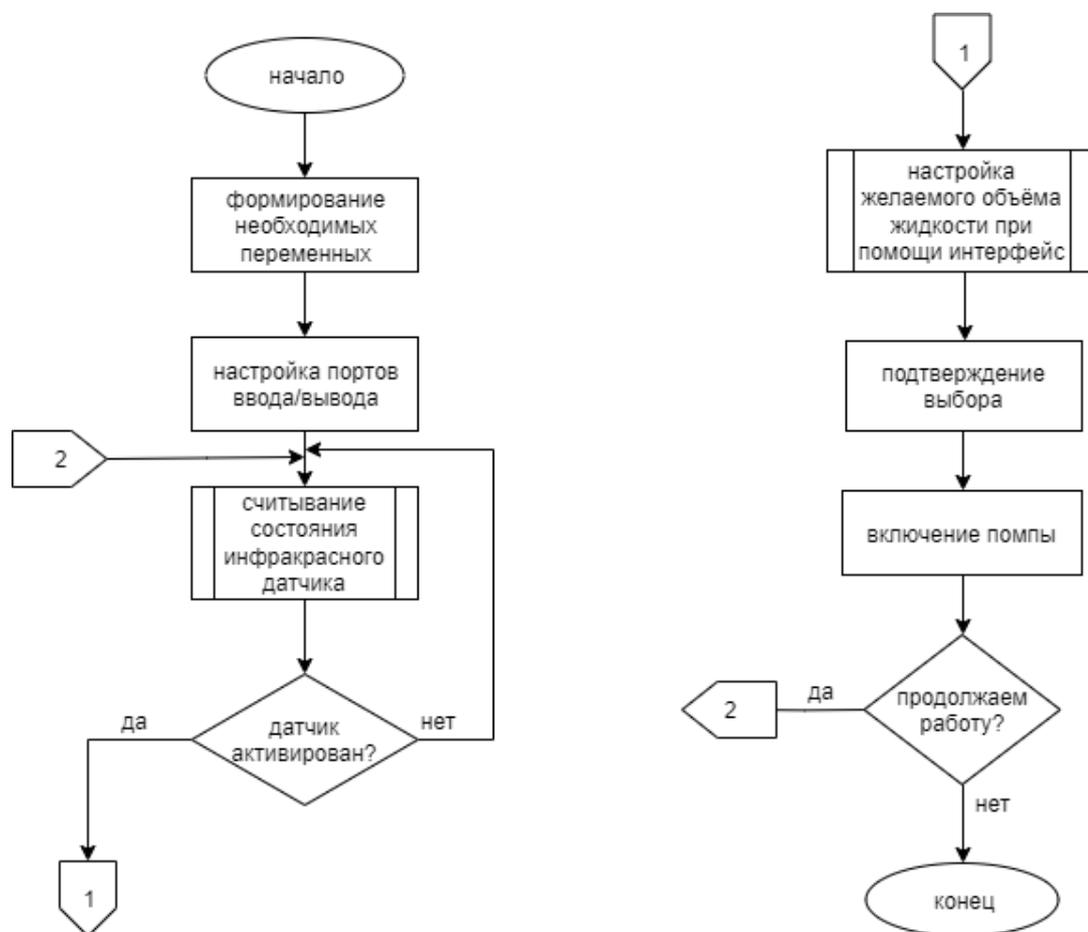


Рисунок 10 – Блок–схема алгоритма работы устройства

В начале алгоритма формируются необходимые переменные, к примеру время, за которое подается определенный объём. Далее настраиваются порты ввода/вывода на микроконтроллере, после чего считывается состояние инфракрасного датчика и если он подает сигнал, то далее, с помощью энкодера и кнопки в интерфейсе устройства выбирается желаемый для подачи объём и включаются помпы. Полученный программный код представлен в приложении А.

3 ОХРАНА ТРУДА И ПРОМЫШЛЕННАЯ ЭКОЛОГИЯ

3.1 Техника безопасности на рабочем месте инженера-электроника

3.1.1 Требования безопасности перед эксплуатацией

3.1.1.1 Инженер-электроник может не приступать к своей работе, если у него есть: сомнения в обеспечении безопасности на рабочем месте при выполнении предстоящей работы.

3.1.1.2 Перед началом работы инженер-электроник должен обратить внимание на рациональную организацию рабочего места, подготовить необходимые инструменты, оборудование и проверить их работу.

3.1.1.3 Перед включением электронных устройств в электросеть инженер-электроник должен также визуально проверить исправность розеток, вилок, электропроводки и кабелей, используемых между машинами, а также соединения для всех устройств.

3.1.1.4 Перед началом работ по техническому обслуживанию и ремонту электронных устройств инженер-электроник должен принять необходимые меры по обеспечению безопасности при работе, в том числе меры электробезопасности, которые включают в себя следующее:

- Выполните необходимые отключения и примите меры для предотвращения неправильного или самопроизвольного включения электронных устройств во время технического обслуживания или ремонта ИТ.

- Проверьте отсутствие напряжения в электронном измерительном приборе.

- При необходимости защитите оставшиеся токопроводящие части напряжение и установите предупреждающие знаки безопасности.

3.1.1.5 Инженер-электроник должен лично убедиться в том, что все меры необходимы для обеспечения безопасности работников; если есть сомнения относительно адекватности и правильности принятых мер и

возможности безопасного выполнения работы, инженер-электроник не должен начинать работу.

3.1.1.6 Инженер-электроник должен проверить наличие и исправность всего необходимого электроразличительного оборудования перед началом работы.

3.1.1.7 Инженеры-электроники должны помнить при работе с персональным компьютером следующее:

- Взаимное расположение компьютеров влияет на уровень электромагнитного излучения, которое они излучают.

- Для обеспечения безопасности расстояние между компьютерами должно быть не менее 1,5 м.

- Персональные компьютеры рекомендуется размещать таким образом, чтобы: левая панель машины была обращена либо к стене, либо к коридору без постоянных рабочих мест.

- Терминалы видеодисплея не должны располагаться так, чтобы экраны были обращены друг к другу.

3.1.2 Требования безопасности при эксплуатации

3.1.2.1 При выполнении работ по техническому обслуживанию, профилактических осмотров и текущего ремонта электронного оборудования инженер-электроник должен соблюдать функциональные документы, в том числе инструкции по применению от производителей.

3.1.2.2 Подключая электронные устройства к электросети, используйте фирменный шнур питания, входящий в комплект оборудования.

3.1.2.3 Во время технического обслуживания и ремонта электронных устройств используйте средства электроразличительной защиты (например, индикатора напряжения, ручного инструмента и т.п.) инженер-электроник не должен приближаться к токоведущим частям расстояние меньше длины изолирующей части этих средств.

3.1.2.4 Отсутствие напряжения должно быть проверено указателем напряжения заводского изготовления, исправность которого необходимо проверить перед использованием.

3.1.2.5 При работе с электронными устройствами ручной инструмент следует использовать только с изолирующими рукоятками; отвертки, кроме того, должны иметь изолированный стержень.

3.1.2.6 Если необходимо выполнять работу без снятия напряжения и при использовании изолирующего защитного оборудования инженер-электроник должен помнить и соблюдать следующие правила техники безопасности:

- Изолирующие части защитного оборудования можно удерживать только ручками за ограничительный круг.

- Изолирующие части защитного оборудования должны располагаться таким образом, чтобы: не было риска перекрытия поверхности изоляции частей, несущих токопроводящие провода, или короткого замыкания на землю.

- Следует использовать только сухие и чистые средства защиты изоляционных деталей с неповрежденным лаком.

- В случае нарушения целостности лакокрасочного покрытия или обнаружения других неисправностей изолирующих частей защитного оборудования их использование прекращается.

3.1.2.7 Чтобы избежать электротравм не допускается оставлять неизолированные концы проводов после разборки блоков электронного оборудования.

3.1.2.8 Если необходимо выполнить пайку с использованием проводов или деталей, оловянно-свинцовые припой типа ПОС, инженер-электроник должен иметь в виду и соблюдать следующие меры предосторожности:

- Поскольку припой типа ПОС содержат свинцовый состав, следует помнить, что при этом одежда, кожа рук загрязняются парами свинца, что может привести (при превышении ПДК) к отравлению свинцом в организме.

- Для предотвращения ожогов лица и глаз от разлетающихся частей расплавленного припоя следует использовать пинцет для поддержки спаянных проводов, деталей.

– Во избежание возгорания следует соблюдать осторожность при работе с электрическим паяльником и использовании специальных подставок.

3.1.2.9 Если во время работы необходимо было использовать ручной электроинструмент, то в этом случае необходимо знать и выполнять следующие основные меры электробезопасности:

– Провода или кабели ручного электроинструмента по возможности должны подвешиваться.

– Предотвращать возможность короткого замыкания, прямого горячего и влажного контакта проводов и кабелей с металлическими поверхностями или предметами не допускается.

– При обнаружении неисправностей, указанных в руководстве по эксплуатации, электроинструмент должен быть немедленно остановлен.

– В случае внезапного отключения электроэнергии во время работы электроинструмента или перерыва он должен быть отключен от сети электропитания.

3.1.2.10 Инженер-электроник должен знать, что рациональная организация рабочего места помогает снизить утомляемость, повысить производительность и снизить вероятность повреждения.

3.1.2.11 Во избежание ухудшения зрения на экране должен быть установлен оптимальный цветовой режим (если это возможно); рекомендуются ненасыщенные цвета: светло-зеленый, желто-зеленый, желто-оранжевый, желто-коричневый; по возможности следует избегать насыщенных цветов, особенно красного, синего, ярко-зеленого.

3.1.2.12 Чтобы уменьшить зрительную и скелетно-мышечную усталость, инженер-электроник должен придерживаться установленного режима работы и отдыха.

3.1.2.13 Для снятия зрительного и позотонического напряжения человек, работающий за персональным компьютером, должен устраивать микропаузы 1-3 минуты во время работы.

3.1.2.14 Для уменьшения негативного эффекта монотонности рекомендуется использовать чередующиеся операции с осмысленным текстом и числовыми данными (изменение содержания работ), чередующееся редактирование текста и ввод данных (изменение содержания работы).

3.1.3 Требования безопасности в аварийных ситуациях

3.1.3.1 При обнаружении нарушений требований охраны труда, которые создают угрозу здоровью или личной безопасности, инженер-электроник должен связаться с мастером и сообщить ему об этом; до устранения угрозы вы должны прекратить работу и покинуть опасную зону.

3.1.3.2 В случае несчастного случая, отравления, внезапного заболевания необходимо оказать пострадавшему немедленную первую помощь, вызвать врача или помочь доставить пострадавшего к врачу, а затем сообщить управляющему зданием о происшествии.

3.1.3.3 При обнаружении пожара или признаков горения (дым, запах гари, повышение температуры и т.д.) необходимо немедленно уведомить службу пожарной безопасности по телефону 01.

3.1.4 Требования безопасности по окончании работы

3.1.4.1 По окончании работ инженер-электроник должен отключить электронное оборудование от сети, привести в порядок рабочее место, убрать съемные магнитные носители, документацию и т.д.

3.2 Опасные производственные факторы на рабочем месте инженера-электроника

Потенциальные опасные и вредные производственные факторы на рабочем месте инженера-электронщика могут включать следующие типы ОВПФ. Опасные и вредные производственные факторы на рабочем месте инженера-электронщика:

Классификация опасных и вредных производственных факторов
Наименование ОВПФ

Физические повышенная температура поверхности оборудования и обрабатываемых материалов;

недостаточное освещение на рабочем месте;

повышенное электромагнитное излучение;

повышенное значение напряжения в цепи, замыкание которой может произойти через тело человека.

Химические повышенная концентрация паров расплавленного олова в воздухе рабочей зоны

Биологические:

Психофизиологические перенапряжение зрительного анализатора

Повышенная температура поверхности оборудования и обрабатываемых материалов. Оборудование и принадлежности, нагретые до температуры пайки, могут вызвать ожоги при случайном прикосновении или при расплавлении материала в процессе пайки. Такие травмы обычно возникают при нарушении технологической системы, при неправильной организации работ по пайке.

Недостаточное освещение на рабочем месте. Освещение производственных помещений вышеупомянутых предприятий оказывает значительное влияние на производительность и качество работ. Неудовлетворительное промышленное освещение приводит к повышенной утомляемости, снижению остроты зрения и, как следствие, снижению качества выполняемых работ, особенно в процессе пайки.

На рабочем месте используется комбинированное освещение.

Электромагнитное излучение. Источниками электромагнитного излучения являются персональный компьютер с системным блоком Intel Pentium и мониторы LG FLATRONL1717S, а также паяльники, используемые в качестве нагревательного оборудования.

Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека. При работе паяльного оборудования используется напряжение 220 В. В случае повреждения

изоляции опасное напряжение человека может возникнуть в непроводящих частях, например, на корпусе устройства, и привести к поражению электрическим током паяльника (ожоги, поражения электрическим током и т.д.).

Повышенная концентрация паров расплавленного олова в воздухе рабочей зоны. Источниками вредных химических веществ, попадающих в воздушную среду, являются припой, изготовленные из олова, и флюсы. Таким образом, олово проникает в организм человека, и при несоблюдении требований безопасности эти вещества могут вызвать профессиональные заболевания или отравления.

Кроме того, при подготовке сборочных единиц к пайке с соединяемых поверхностей удаляются оксидные пленки, масла и другие загрязнения, что достигается механическим отслаиванием, обезжириванием, химическим травлением. Растворы для обезжиривания и травления часто содержат вещества, которые, попадая в организм человека, могут вызвать отравление или ожоги.

3.3 Методы утилизации вычислительной техники

Теоретически, есть три способа решить эту проблему: продать, передать или отменить и утилизировать оборудование. Продать - этот вариант верен только в том случае, если есть покупатель и ремонтпригодность самого оборудования. Но обычно списывают только самые ненужные из них.

Продажа или передача старого оборудования школе или детскому саду также не всегда является лучшим вариантом. Качество старого оборудования довольно низкое, и во многих случаях его больше нельзя использовать нормально.

Есть еще вариант - утилизация компьютерной и оргтехники. Но организация должна провести вывоз оборудования, подтвердив факт передачи компьютеров и оргтехники.

Стандартный процесс утилизации списанного и компьютерного оборудования включает в себя следующие этапы:

– Компания, целью которой является списания устройств, создаст специальную внутреннюю комиссию. Его главная задача - принять коллегиальное решение о типе оборудования, которое пора списывать.

– Это решение комиссии о выводе из эксплуатации компьютеров и оргтехники обязательно должно основываться на экспертном заключении. Эксперт может быть либо штатным сотрудником компании, либо независимым экспертом, привлеченным извне. Вам определенно нужен производитель компьютерной или офисной техники или другая компания с лицензией на обслуживание и ремонт этого оборудования. Такой акт технической экспертизы компьютеров и оборудования документирует, что оборудование неисправно, его ремонт нецелесообразен и настало время для его выхода на пенсию. Старое оборудование может быть списано и утилизировано.

– Для того, чтобы вывести из эксплуатации офисное оборудование и компьютеры служба, которых окончательно завершена и забыта, вам также необходимо предоставить документальные доказательства того, что они были утилизированы надлежащим образом, без дальнейшего разрушения нашей экосистемы, и что они были разложены на тяжелые металлы и токсичные соединения.

3.4 Вывод по разделу

В разделе охрана труда и промышленная экология были рассмотрены темы:

- техника безопасности на рабочем месте инженера-электроника;
- опасные производственные факторы на рабочем месте инженера-электроника;
- методы утилизации вычислительной техники.

По итогам подраздела методы утилизации вычислительной техники было выявлено, что при утилизации отходов вычислительной техники необходимо пользоваться услугами специальных организаций, выполняющих утилизацию, в соответствии с всеми нормами и правилами. Утилизация необходима по причине минимизации ущерба экологии от отходов вычислительной техники, а также по причине извлечения драгоценных металлов из старых компонентов вычислительной техники.

В подразделе опасные производственные факторы на рабочем месте инженера-электроника были рассмотрены сами понятия вредных и опасных факторов на рабочем месте инженера-электроника, а также понятия, близко связанные с безопасностью труда. Рассмотрена классификация негативных факторов, с подробным рассмотрением каждого пункта и приведением примером негативных факторов.

В подразделе техника безопасности на рабочем месте инженера-электроника были описаны и изучены требования безопасности перед эксплуатацией, во время эксплуатации и требования безопасно в аварийных ситуациях.

4 ОРГАНИЗАЦИОННО–ЭКОНОМИЧЕСКАЯ ЧАСТЬ

4.1 Расчет затрат на разработку программы для микроконтроллера

Затраты на разработку программы для микроконтроллера рассчитываются по формуле 1.

$$Z_{РПР} = Z_{ФОТР} + Z_{ОВФ} + Z_{ЭВМ} + Z_{СПП} + Z_{ХОН} + P_H, \quad (1)$$

где $Z_{ФОТР}$ – общий фонд оплаты труда разработчиков программы,

$Z_{ОВФ}$ – начисления на заработную плату разработчиков программы во внебюджетные фонды,

$Z_{ЭВМ}$ – затраты, связанные с эксплуатацией техники,

$Z_{СПП}$ – затраты на специальные программные продукты, необходимые для разработки программы для микроконтроллера,

$Z_{ХОН}$ – затраты на хозяйственно-операционные нужды (бумага, литература, носители информации и т.п.),

P_H – накладные расходы ($P_H = 30\%$ от $Z_{ФОТР}$).

При разработке программы для микроконтроллера общее время разработки составило 1 месяца.

Фонд оплаты труда за время работы над программой для микроконтроллера рассчитывается по формуле 2.

$$Z_{ФОТР} = \sum_{j=1}^m O_{Pj} \cdot T_{РПРj} \cdot (1 + k_D)(1 + k_Y), \quad (2)$$

где O_{Pj} – оклад j -го разработчика. В разработке участвовал 1 человек, его оклад составляет 17500 руб.,

$T_{РПРj}$ – общее время работы над программой в месяцах, $T_{РПР} = 1$,

k_D – коэффициент дополнительной зарплаты, $k_D = 20\% = 0,2$,

k_Y – районный коэффициент, $k_Y = 0,15$.

Таким образом,

$$Z_{ФОТР} = 17500 \cdot 1 \cdot (1 + 0,2) \cdot (1 + 0,15) = 24150 \text{ руб.}$$

Страховой взнос во внебюджетные фонды складываются из взносов на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний.

Страховые взносы складываются из обязательного пенсионного страхования (ОПС), отчислений в фонд социального страхования и отчислений в фонд обязательного медицинского страхования.

Значения всех используемых ставок приведены в таблице 6.

Таблица 6 – Значения ставок страховых взносов

№	Наименование внебюджетного фонда	Размер ставок, %
1	Пенсионный фонд	22
2	Фонд социального страхования	2,9
3	Федеральный фонд обязательного медицинского страхования	5,1
	Итого:	30

Сумма начислений на заработную плату во внебюджетные фонды рассчитывается по формуле 3.

$$Z_{ОВФ} = 0,3 \cdot Z_{ФОТР}, \quad (3)$$

$$Z_{ОВФ} = 0,3 \cdot 24150 = 7245 \text{ руб.}$$

Затраты, связанные с использованием вычислительной и оргтехники рассчитываются по формуле 4.

$$Z_{ЭВМ} = T_{МРПР} \cdot k_{Г} \cdot n \cdot C_{М-ч} \quad (4)$$

где $k_{Г}$ – коэффициент готовности ЭВМ, $k_{Г} = 0,95$,

n – количество единиц техники, равно 1,

$C_{М-ч}$ – себестоимость машино–часа, $C_{М-ч} = 6$ руб.,

$T_{МРПР}$ – машинное время работы над программой, равно 1 мес.

Перевод рабочего времени в часы осуществляется по формуле 5.

$$T_{час} = T_{мес} \cdot Ч_{РД} \cdot T_{см} \cdot K_{см}, \quad (5)$$

где $T_{час}$ – рабочее время, ч,

$T_{мес}$ – рабочее время, мес, ($T_{мес} = 1$),

$Ч_{РД}$ – число рабочих дней, ($Ч_{РД} = 22$),

$T_{см}$ – продолжительность рабочей смены, ($T_{см} = 8$ ч),

$K_{см}$ – количество рабочих смен, ($K_{см} = 1$).

Таким образом, время на разработку программы для микроконтроллера с использованием ЭВМ составляет:

$$T_{\text{час}} = 1 \cdot 22 \cdot 8 \cdot 1 = 176 \text{ часа,}$$

$$Z_{\text{ЭВМ}} = 176 \cdot 0,95 \cdot 1 \cdot 6 = 1003,2 \text{ руб.}$$

Затраты на специальные программные продукты, необходимые для разработки программы для микроконтроллера рассчитываются по формуле 6.

$$Z_{\text{СПП}} = \sum_{p=1}^n C_p \quad (6)$$

где C_p – цена p -го специального программного продукта.

Перечень программных продуктов специального назначения приведен в таблице 7.

Таблица 7 – Программные продукты специального назначения

№	Название ПП	Цена, руб.
1	Среда разработки Arduino IDE	0
2	easyEDA	0
	Итого:	0

Использованные программные продукты бесплатны, поэтому:

$$Z_{\text{СПП}} = 0 \text{ руб.}$$

Затраты на хозяйственно–организационные нужды приведены в таблице 8 и вычисляются по формуле 7.

$$Z_{\text{ХОИ}} = \sum_{\tau=1}^n C_{\tau} \cdot K_{\tau}, \quad (7)$$

где C_{τ} – цена τ -го товара, руб.,

K_{τ} – количество τ -го товара.

Таблица 8 – Затраты на хозяйственно–организационные нужды

Наименование	Цена за единицу (руб.)	Кол-во (шт.)	Всего (руб.)
USB–флеш–накопитель	350	1	350
Бумага	2	100	200
		Итого:	550

$$Z_{\text{ХОИ}} = 350 \cdot 1 + 2 \cdot 100 = 550 \text{ руб.}$$

Накладные расходы рассчитываются по формуле 8.

$$P_H = Z_{\text{ФОТР}} \cdot k_{\text{НР}}, \quad (8)$$

$$P_H = 24150 \cdot 0,3 = 7245 \text{ руб.}$$

Таким образом, затраты на разработку программы для микроконтроллера, рассчитанные по формуле 1 составят:

$$Z_{\text{РПР}} = 24150 + 7245 + 1003,2 + 0 + 550 + 7245 = 40193,2 \text{ руб.}$$

4.2 Расчет затрат на внедрение программы для микроконтроллера

Затраты на внедрение программы для микроконтроллера ($Z_{\text{ВПР}}$) рассчитываются по формуле 9.

$$Z_{\text{ВПР}} = Z_M + Z_{\text{КТС}} \cdot (1 + k_{\text{тун}}) + Z_{\text{ПО}} + Z_{\text{ФОТВ}} + Z_{\text{ОФВ}} + Z_{\text{ЭВМ}} + P_{\text{ком}} + P_H, \quad (9)$$

где Z_M – затраты на приобретение материалов, руб.,

$Z_{\text{КТС}}$ – затраты на приобретение комплекса технических средств, руб.,

$Z_{\text{ПО}}$ – затраты на приобретение программного обеспечения (включают стоимость разработанной программы, а также других существующих ПП, необходимых для функционирования системы), руб.,

$Z_{\text{ФОТВ}}$ – затраты на оплату труда работников, занятых внедрением проекта, руб.,

$Z_{\text{ОФВ}}$ – страховой взнос во внебюджетные фонды с заработной платы работников, занятых внедрением проекта, руб.,

$Z_{\text{ЭВМ}}$ – затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения, руб.,

$P_{\text{ком}}$ – командировочные расходы, руб.,

P_H – накладные расходы, руб.,

$k_{\text{тун}}$ – коэффициент транспортирования, установки и наладки комплекса технических средств, определяется действующими нормативами организации, а также спецификой конкретного проекта.

Затраты на приобретение материалов (Z_M) приведены в таблице 9.

Таблица 9 – Затраты на приобретение материалов

Наименование	Цена за единицу (руб.)	Кол-во (шт.)	Всего (руб.)
Arduino NANO v3	200	1	200
Энкодер WAVGAT 5V	86	1	86
Кнопка Переключатель pbs33b 12V	30	1	30
Дисплей LCD 1602	92	1	92
Модуль гнездового преобразователя MICRO USB/DIP	43	1	43
Драйвер MX1508	55	1	55
Инфракрасный датчик	33	1	33
Мембранная пищевая помпа для жидкости DC12V	734	2	1468
Внешний аккумулятор (повербанк)	850	1	850
Конденсатор 47uF	10	1	10
Медные провода	2,65	20	53
Пластиковый корпус из хозяйственного магазина	120	1	120
Второе дно для корпуса	55	1	55
Трубка ПВХ пищевая	300	1	300
Итого:			3395

Дополнительного приобретения компьютеров или других КТС не требуется, следовательно, $Z_{\text{КТС}} = 0$.

Затраты на приобретение программного обеспечения в данном случае равны затратам на разработку и составляют $Z_{\text{ПО}} = 40193,2$ руб.

Внедрением занят один системный инженер с окладом 18000 руб. Время внедрения – 0,25 месяца. По формуле рассчитываем затраты на оплату труда и страховой взнос во внебюджетные фонды.

$$Z_{\text{ФОТВ}} = 18000 \cdot 0,25 \cdot (1 + 0,15) = 5175 \text{ руб.}$$

$$Z_{\text{ОВФ}} = 5175 \cdot 0,3 = 1552,5 \text{ руб.}$$

Затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения, составят:

$$Z_{\text{ЭВМ}} = 0,25 \cdot 22 \cdot 8 \cdot 6 = 264 \text{ руб.}$$

Командировочные расходы при внедрении программы для микроконтроллера не планируются, следовательно, $R_{\text{ком}} = 0$.

Так как коэффициент накладных расходов по данным организации составляет $k_{НР} = 0,3$, то величина накладных расходов равна 1552.5 руб.

Суммарные затраты на внедрение составят:

$$Z_{ВПР} = 3395 + 0 + 40193,2 + 5175 + 1552,5 + 264 + 0 + 1552,5 = 52132,2 \text{ руб.}$$

4.3 Расчет эксплуатационных текущих затрат по программе для микроконтроллера

Годовые затраты на обработку результатов до внедрения разработанной программы для микроконтроллера рассчитываются по формуле 10.

$$C_1 = 3П_1 + ОТ_{ВН1} + З_{ЭВМ1} + М_{з1} + НР_1, \quad (10)$$

где $3П_1$ – затраты на оплату труда сотрудника на выполнение функций до внедрения проектного решения,

$ОТ_{ВН1}$ – страховой взнос во внебюджетные фонды,

$З_{ЭВМ1}$ – затраты, связанные с эксплуатацией ЭВМ,

$М_{з1}$ – годовые материальные затраты на сопровождение программы для микроконтроллера составляют 2200 руб.,

$НР_1$ – накладные расходы.

Временные затраты работы сотрудника в месяцах рассчитываются по формуле 11.

$$T_{1мес} = \frac{T_{1час}}{Ч_{рд} \cdot Ч_{рч}}, \quad (11)$$

где $T_{1мес}$, $T_{1час}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{1час} = 275$ часов),

$Ч_{рд}$ – число рабочих дней в месяц,

$Ч_{рч}$ – число рабочих часов в день.

$$T_{1мес} = \frac{275}{22 \cdot 8} = 1,6 \text{ мес.}$$

Тогда затраты на оплату труда сотрудника рассчитываются по формуле 12.

$$ЗП_1 = O_c \cdot T_{1\text{мес}} \cdot (1 + k_d) \cdot (1 + k_y), \quad (12)$$

где O_c – оклад сотрудника (оклад составляет 20000 руб.),

$$ЗП_1 = 20000 \cdot 1,6 \cdot (1 + 0,2) \cdot (1 + 0,15) = 44160 \text{ руб.}$$

Страховой взнос до внедрения вычисляют по формуле 13.

$$ОТ_{ВН1} = ЗП_1 \cdot 0,3, \quad (13)$$

$$ОТ_{ВН1} = 44160 \cdot 0,3 = 13248 \text{ руб.}$$

Рассчитываем затраты, связанные с эксплуатацией ЭВМ до внедрения по формуле 14.

$$З_{ЭВМ1} = T_{1\text{час}} \cdot C_{M-ч}, \quad (14)$$

$$З_{ЭВМ1} = 275 \cdot 6 = 1650 \text{ руб.}$$

Подставив соответствующие значения в формулу 10, получим:

$$C_1 = 44160 + 13248 + 1650 + 2200 = 61258 \text{ руб.}$$

Годовые затраты на эксплуатацию системы после внедрения программы для микроконтроллера рассчитываются аналогично по формуле 15.

$$C_2 = ЗП_2 + ОТ_{ВН2} + З_{ЭВМ2} + M_{з2} + НР_2, \quad (15)$$

где $ЗП_2$ – затраты на оплату труда сотрудника после внедрения,

$ОТ_{ВН2}$ – страховой взнос во внебюджетные фонды,

$З_{ЭВМ2}$ – затраты, связанные с эксплуатацией ЭВМ после внедрения,

$M_{з2}$ – материальные затраты, годовые материальные затраты на сопровождение программы для микроконтроллера составляют 2500 руб.,

$НР_2$ – накладные расходы.

Временные затраты работы сотрудника в месяцах вычисляются по формуле 16.

$$T_{2\text{мес}} = \frac{T_{2\text{час}}}{Ч_{рд} \cdot Ч_{рч}}, \quad (16)$$

где $T_{2\text{мес}}$, $T_{2\text{час}}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{2\text{час}} = 50$ часов),

$Ч_{рд}$ – число рабочих дней в месяц,

$Ч_{рч}$ – число рабочих часов в день.

$$T_{2\text{мес}} = \frac{120}{22 \cdot 8} = 0,7 \text{ мес.}$$

Тогда затраты на оплату труда сотрудника высчитываются по формуле 17.

$$ЗП_2 = O_c \cdot T_{2\text{мес}} \cdot (1 + k_d) \cdot (1 + k_y), \quad (17)$$

где O_c – оклад сотрудника (оклад составляет 20000 руб.)

$$ЗП_2 = 20000 \cdot 0,7 \cdot (1 + 0,2) \cdot (1 + 0,15) = 19320 \text{ руб.}$$

Страховой взнос после внедрения вычисляются по формуле 18.

$$O_{T_{ВН2}} = ЗП_2 \cdot 0,3, \quad (18)$$

$$O_{T_{ВН2}} = 19320 \cdot 0,3 = 5796 \text{ руб.}$$

Рассчитываем затраты, связанные с эксплуатацией ЭВМ после внедрения по формуле 19.

$$З_{ЭВМ2} = T_{2\text{час}} \cdot C_{M-Ч}, \quad (19)$$

$$З_{ЭВМ2} = 120 \cdot 6 = 720 \text{ руб.}$$

Подставив соответствующие значения в формулу, получим:

$$C_2 = 19320 + 5796 + 720 + 2500 = 28336 \text{ руб.}$$

Таким образом, текущие затраты на содержание системы до внедрения разработанной программы для микроконтроллера составляют 61258 руб., после внедрения 28336 руб.

4.4 Расчет экономической целесообразности разработки и внедрения информационных технологий

Для разрабатываемого проекта расчет экономической эффективности производится исходя из следующих условий:

– годовые текущие затраты до внедрения автоматизированной системы, $C_1 = 61258$ руб.,

– годовые текущие затраты после внедрения системы, $C_2 = 28336$ руб.,

– горизонт расчета принимается исходя из срока использования разработки, $T = T_n = 2$ годам,

– шаг расчета равен одному году, $t = 1$ году,

– капитальные вложения равны затратам на создание системы, $K = 52132,2$ руб.,

– норма дисконта равна норме дохода на капитал, $E = 12\%$.

Ожидаемая условно-годовая экономия от внедрения системы рассчитывается по формуле 20.

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 + \sum \mathcal{E}_i, \quad (20)$$

где $\mathcal{E}_{\text{уг}}$ – ожидаемая условно-годовая экономия, руб.,

C_1 – годовые текущие затраты до внедрения автоматизированной системы, руб.,

C_2 – годовые текущие затраты после внедрения системы, руб.,

\mathcal{E}_i – ожидаемый дополнительный эффект от различных факторов, руб.

Так как основным фактором, по которому производится расчет экономической эффективности от внедрения программы для микроконтроллера, является уменьшение времени обработки результатов тестирования и дополнительный эффект не учитывается, то $\sum \mathcal{E}_i = 0$.

Подставив вычисленные выше значения в формулу, получим:

$$\mathcal{E}_{\text{уг}} = 61258 - 28336 = 32922 \text{ руб.}$$

где $\mathcal{E}_{\text{уг}}$ – ожидаемая условно-годовая экономия, руб.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле 21.

$$\mathcal{E}_{\text{г}} = \mathcal{E}_{\text{уг}} - K \cdot E_{\text{н}}, \quad (21)$$

где $\mathcal{E}_{\text{г}}$ – ожидаемый годовой экономический эффект, руб.,

$\mathcal{E}_{\text{уг}}$ – ожидаемая условно-годовая экономия, руб.,

K – капитальные вложения (равны затратам на создание ИС), руб.,

$E_{\text{н}}$ – нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле 22.

$$E_n = \frac{1}{T_n}, \quad (22)$$

где T_n – нормативный срок окупаемости капитальных вложений, лет.

Подставив вычисленные выше значения в формулу, получим:

$$E_n = 32922 - 52132,2 \cdot 0,5 = 6855,9 \text{ руб.}$$

Расчетный коэффициент экономической эффективности капитальных вложений рассчитывается по формуле 23.

$$E_p = \frac{\Delta_{уг}}{K}, \quad (23)$$

где E_p – расчетный коэффициент экономической эффективности капитальных вложений,

$\Delta_{уг}$ – ожидаемая условно–годовая экономия, руб.,

K – капитальные вложения на создание системы, руб.

Подставив вычисленные выше значения в формулу, получим:

$$E_p = \frac{32922}{52132,2} = 0,6$$

Расчетный срок окупаемости капитальных вложений определяется по формуле 24.

$$T_p = \frac{1}{E_p}, \quad (24)$$

где E_p – коэффициент экономической эффективности капитальных вложений.

Подставив вычисленные выше значения в формулу, получим:

$$T_p = \frac{1}{0,6} = 1,6 \text{ год.}$$

Срок окупаемости без дисконтирования 1,6 год.

Чистый дисконтированный доход (ЧДД) определяется как сумма текущих эффектов за весь расчетный период, приведенная к начальному шагу, или как превышение интегральных результатов над интегральными затратами.

Если в течение расчетного периода не происходит инфляционного изменения цен или расчет производится в базовых ценах, то величина ЧДД для постоянной нормы дисконта вычисляется по формуле 25.

$$\text{ЧДД} = \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1 + E)^t} - K, \quad (25)$$

где P_t – ожидаемые результаты от внедрения предложенной ИС, руб.,

Z_t – ожидаемые затраты (капитальные и текущие) на создание и эксплуатацию ИС, руб.,

$\Delta t = (P_t - Z_t)$ – эффект, достигаемый на t -м шаге расчета,

K – капитальные вложения,

t – номер шага расчета ($t = 1, 2$),

T – горизонт расчета,

E – постоянная норма дисконта, 12%.

$\Delta t = (P_t - Z_t) = \Delta \text{уг} = 32922$ руб. В том случае, если текущие затраты (Z_t) на весь срок использования разработки равны 0.

$t = 1, 2$ год, т.к. предполагается, что результат от внедрения предложенной ИС будет с текущего года внедрения ИС.

Если ЧДД инвестиционного проекта положителен, то проект является эффективным (при данной норме дисконта).

Тогда суммарный чистый дисконтированный доход за весь горизонт расчета равен:

$$\text{ЧДД} = \Delta_1 \cdot \frac{1}{(1 + E)} + \Delta_2 \cdot \frac{1}{(1 + E)^2}, \quad (26)$$

$$\text{ЧДД} = \frac{32922}{(1 + 0,12)} + \frac{32922}{(1 + 0,12)^2} - 52132,2 = 3507,65 \text{ руб.}$$

Положительное значение чистого дисконтированного дохода, $\text{ЧДД} > 0$, свидетельствует о том, что инвестирование целесообразно и данная ИС может приносить прибыль в установленном объеме.

Индекс доходности (ИД) представляет собой отношение суммы приведенных эффектов к величине капитальных вложений и определяется по формуле 27.

$$\text{ИД} = \frac{1}{K} \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1 + E)^t}, \quad (27)$$

где K – величина капиталовложений или стоимость инвестиций.

$$\text{ИД} = \frac{55639,85}{52132,2} = 1,06$$

Инвестиции считаются эффективными, если индекс доходности выше единицы, $\text{ИД} > 1$, следовательно, инвестиции в данную ИС, эффективны.

Внутренняя норма доходности (ВНД):

при $E_1 \rightarrow \text{ЧДД}_1 > 0$

$E_2 \rightarrow \text{ЧДД}_2 < 0$

$$\text{ВНД} = E_1 + \frac{\text{ЧДД}_1}{\text{ЧДД}_1 - \text{ЧДД}_2} \cdot (E_2 - E_1), \quad (28)$$

при $E_1 \rightarrow \text{ЧДД}_1 > 0$

$E_2 \rightarrow \text{ЧДД}_2 > 0$

$$\text{ВНД} = E_1 + \frac{\text{ЧДД}_1}{\text{ЧДД}_1 + \text{ЧДД}_2} \cdot (E_2 - E_1) \quad (29)$$

$E_1 = 0,11$

$$\text{ЧДД}_1 = \frac{32922}{(1 + 0,11)} + \frac{32922}{(1 + 0,11)^2} - 52132,2 = 4247 \text{ руб.}$$

$E_1 = 0,13$

$$\text{ЧДД}_2 = \frac{32922}{(1 + 0,13)} + \frac{32922}{(1 + 0,13)^2} - 52132,2 = 2785 \text{ руб.}$$

$E_1 \rightarrow \text{ЧДД}_1 > 0$

$E_2 \rightarrow \text{ЧДД}_2 > 0$

$$\text{ВНД} = 0,11 + \frac{4247}{4247 + 2785} \cdot (0,13 - 0,11) = 0,12.$$

Таким образом, норма дисконта должна быть в пределах 11% – 13%.

Показатели экономической целесообразности разработки и внедрения программы для микроконтроллера сведены в таблицу 10.

Таблица 10 – Показатели экономической целесообразности разработки и внедрения программы для микроконтроллера

Наименование показателя	Значения
Затраты на разработку и внедрение ПП, руб.	52132,2
Ожидаемая экономия от внедрения ПП, руб.	32922
Чистый дисконтированный доход, руб.	3507,65
Индекс доходности	1,06
Внутренняя норма доходности	0,12
Дисконтированный срок окупаемости, лет	1,6
Срок морального старения, года	2

Произведенные расчеты свидетельствуют, что внедрение, разработанной в ВКР программы для микроконтроллера, позволит сократить временные затраты на обработку результатов тестирования, что приведет к сокращению годовых текущих затрат на 32922 рублей

Опираясь на оценку экономической эффективности, можно сделать вывод о том, что разработка и внедрение предлагаемой программы для микроконтроллера является экономически обоснованной и целесообразной.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель. Была создана структурная схема автоматизированного устройства дозирования жидкостей. Был создан макет, по средствам имеющихся компонентов и материалов, который способен подключаться к компьютеру. Данная разработка в дальнейшем может быть модернизирована в зависимости от поставленных задач. Был спроектирован корпус устройства и рассчитаны нужные размеры

В ходе выпускной квалификационной работы были изучены программные интерфейсы для написания кода. А также были изучены программы для построения чертежей и принципиальной схемы. Получены навыки процесса сборки, правильного подключения элементов друг к другу, начального уровня программирования.

Разработанная система является основой, способной принимать множество трансформаций и модернизаций, различными путями, чтобы удовлетворить современным нуждам заказчика, при этом не обязательно использовать компоненты и программы, представленные в данной работе, так как существует множество вариаций и каждый может выбрать то, что необходимо конкретно для выполнения поставленных задач, при этом основная идея устройства сохраняется. К примеру, устройство можно модернизировать для нескольких ёмкостей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Как работают автоматические дозаторы розлива жидкостей и паст? [электронный ресурс]. URL: <https://dozaplus.ru/sposoby-dozirovania-gidkostey> – (дата обращения 07.06.2022 г.)
2. Как подобрать дозатор жидкости? [электронный ресурс]. URL: <https://dozaplus.ru/pravilniy-podbor-dozatora> – (дата обращений 08.06.2022 г.)
3. Виды молочного сырья и показатели его качества (страница 10) – физико-химические свойства молока [электронный ресурс]. URL: <https://en.ppt-online.org/611418> – (дата обращения 06.08.2022 г.)
4. Техника и технологии производства и переработки растительных масел (страница 27) – физико-химические показатели растительных масел [электронный ресурс] URL: <https://tstu.ru/book/elib/pdf/2010/tarov.pdf> – (дата обращения 06.08.2022 г.)
5. Теплофизические характеристики пюре, повидло, соков [электронный ресурс]. URL: <http://thermalinfo.ru/svoystva-produktov/ovoshhi-i-frukty/tepfizicheskie-harakteristiki-pyure-povidlo-sokov> – (дата обращения 06.08.2022 г.)
6. Качественный контроль фруктового сока [электронный ресурс]. URL: <https://paar.ru/upload/iblock/63c/63c72bcd350f402ee3bcdd4fa8542515.pdf> – (дата обращения 06.08.2022 г.)
7. Изобарический розлив [электронный ресурс]. URL: <http://rozliv-info.ru/izobaricheskiy-rozliv> – (дата обращения 08.03.2022 г.)
8. О бармене роботе: функции, цена, где купить [электронный ресурс]. URL: <https://motocarrello.ru/jelektrotehnologii/roboty/1693-barmen-robot.html> – (дата обращения 09.03.2022 г.)
9. Домашний бар Drinkworks от Keurig - Приготовьте качественный коктейль [электронный ресурс]. URL: <https://advancedmixology.com/blogs/art-of-mixology/drinkworks-home-bar-by-keurig> – (дата обращения 09.03.2022 г.)
10. Домашний бар Drinkworks смешивает коктейли на основе капсул от AB InBev и Keurig [электронный ресурс]. URL:

<https://www.cnet.com/reviews/drinkworks-home-bar-by-keurig-preview> – (дата обращения 09.03.2022 г.)

11. Домашний бар Drinkworks от KeurigReview [электронный ресурс]. URL: <https://elitechs.ru/domashnij-bar-drinkworks-ot-keurig-review> – (дата обращения 09.03.2022 г.)

12. Робот бармен на разлив [электронный ресурс]. URL: <http://batterylow.ru/nalivator-alkogolja-s-radio> – (дата обращения 09.03.2022 г.)

13. Автоматический бармен «Barobot» [электронный ресурс]. URL: <https://www.robot96.ru/catalog/exclusive/barobot> – (дата обращения 09.03.2022 г.)

14. Автоматический наливатор напитков [электронный ресурс] URL: <https://runprod.ru/arduino/nalivator/> – (дата обращения 09.03.2022 г.)

15. Собираем наливатор по схеме [электронный ресурс] URL: <https://compacttool.ru/practic.php?view=17> – (дата обращения 09.03.2022 г.)

16. Драйвер моторов MX1508 [электронный ресурс] URL: <https://duino.ru/mx1508-drayver-dvigatelay>

17. Список микроконтроллеров ATmega. [электронный ресурс]. URL: <https://chipinfo.pro/mcu/avr/atmegalist.shtml> – (дата обращения 09.03.2022 г.)

18. Баев Б.П. Микропроцессорные системы бытовой техники Учебник для вузов. — 2-е изд., испр. и доп. — М.: Горячая линия-Телеком, 2005. — 480 с.

19. Артеменко С.В., Тихоненко Д.В. Утилизация компьютерной техники. 2020. №8. URL: <https://cyberleninka.ru/article/n/utilizatsiya-kompyuternoj-tehniki> (дата обращения: 25.04.2022)

20. Идентификация опасных и вредных производственных факторов на рабочем месте инженера-электронщика [электронный ресурс] URL: https://studbooks.net/1510896/bzhd/identifikatsiya_opasnyh_vrednyh_proizvodstvennyh_faktorov_rabochem_meste_inzhenera_elektronschika - (дата обращения 23.04.2022г).

Приложение А – Программный код

```
// НАСТРОЙКИ//
#define NUM_SHOTS 1 // количество ёмкостей
#define TIMEOUT_OFF 5 // таймаут на выключение (перестает дёргать привод),
минут
// положение серво над центрами рюмок
const byte shotPos[] = {36, 72, 108, 144};
// время заполнения 50 мл
const int time50ml = 5500;
float currentVolume;
int totalRumok = 0;
int totalGramm = 0;
int posInfo;
#define KEEP_POWER 0 // 1 - система поддержания питания ПБ, чтобы он не спал
// отладка
#define DEBUG_UART 1
// ПИНЫ//
#define PUMP1_POWER 3
#define PUMP2_POWER 2
#define BTN_PIN 7
#define ENC_SW 8
#define ENC_DT 9
#define ENC_CLK 10
#define DISP_DIO 11
#define DISP_CLK 12
const byte SW_pins[] = {A0, A1, A2, A3};

// БИБЛИОТЕКИ//
#include <TM1637.h>
#include <microLED.h>
#include "encUniversalMinim.h"
#include "buttonMinim.h"
#include "timer2Minim.h"
#include <LCD_1602_RUS.h>
#include <Wire.h>
```

```

#include <EEPROM.h>
LCD_1602_RUS lcd(0x27, 16, 2);
// ===== ДАТА =====
#define ORDER_GRB          // порядок цветов ORDER_GRB / ORDER_RGB /
ORDER_BRG
#define COLOR_DEPTH 2 // цветовая глубина: 1, 2, 3 (в байтах)
LEDdata leds[NUM_SHOTS]; // буфер ленты типа LEDdata (размер зависит от
COLOR_DEPTH)
TM1637 disp(DISP_CLK, DISP_DIO);
// пин clk, пин dt, пин sw, направление (0/1), тип (0/1)
encMinim enc(ENC_CLK, ENC_DT, ENC_SW, 1, 1);
buttonMinim btn(BTN_PIN);
buttonMinim encBtn(ENC_SW);
timerMinim LEDtimer(100);
timerMinim FLOWdebounce(20);
timerMinim FLOWtimer(2000);
timerMinim WAITtimer(300);
timerMinim TIMEOUTtimer(15000); // таймаут дёргания приводом
timerMinim POWEROFFtimer(TIMEOUT_OFF * 60000L);
bool LEDchanged = false;
bool pumping = false;
int8_t curPumping = -1;
enum {NO_GLASS, EMPTY, IN_PROCESS, READY} shotStates[NUM_SHOTS];
enum {SEARCH, MOVING, WAIT, PUMPING} systemState;
bool workMode = false; // 0 manual, 1 auto
int thisVolume = 50;
bool systemON = false;
bool timeoutState = false;
//МАКРО//
#define pumpON() digitalWrite(PUMP1_POWER, 0)
#define pumpOFF() digitalWrite(PUMP1_POWER, 1)
#define pumpON() digitalWrite(PUMP2_POWER, 0)
#define pumpOFF() digitalWrite(PUMP2_POWER, 1)
#if (DEBUG_UART == 1)
#define DEBUG(x) Serial.println(x)

```

```

#else
#define DEBUG(x)
#endif

void setup() {
  Serial.begin(9600);
  Serial.println("Start");
  //Инициализируем тотальные значение!
  /*
  writeValEeprom(0, 1); // Сбрасываем счетчик включений (хранится в ячейке 1
EEPROM)
  writeValEeprom(0, 3); //Сбрасываем счетчик рюмок (хранится в ячейке 3
EEPROM)
  writeValEeprom(0, 5); //Сбрасываем счетчик грамм (хранится в ячейке 5
EEPROM)
  */
  //Считываем последний запомненный объем порции в граммах (хранится в ячейке
7 EEPROM)
  //Увеличиваем счетчик включения
  // тыкаем ленту
  // настройка пинов
  pinMode(PUMP1_POWER, 1);
  for (byte i = 0; i < NUM_SHOTS; i++) {
    pinMode(SW_pins[i], INPUT_PULLUP);
  }
  pinMode(PUMP2_POWER, 1);
  for (byte i = 0; i < NUM_SHOTS; i++) {
    pinMode(SW_pins[i], INPUT_PULLUP);
  }
  }
  // старт дисплея
  disp.clear();
  disp.brightness(7);
  //настройка серво
  delay(800);
  lcd.init(); // initialize the lcd
  lcd.backlight();
  lcd.setCursor(0, 0);

```

```

lcd.print(" ПРИВЕТ!!! ");
lcd.setCursor(0, 1);
lcd.print(" КАК ДЕЛА? ");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.setCursor(0, 1);
delay(3500);
lcd.clear();
}
// луп
void loop() {
  encTick();
  btnTick();
  flowTick();
  LEDtick();
  timeoutTick();
  // Serial.println(readValEeprom(7));
}
// мини-класс для работы с кнопкой, версия 1.0
#pragma pack(push,1)
typedef struct {
  bool holdedFlag: 1;
  bool btnFlag: 1;
  bool pressF: 1;
  bool clickF: 1;
  bool holdF: 1;
} buttonMinimFlags;
#pragma pack(pop)
class buttonMinim {
public:
  buttonMinim(uint8_t pin);
  boolean pressed();
  boolean clicked();
  boolean holding();

```

```

    boolean holded();
private:
    buttonMinimFlags flags;
    void tick();
    uint32_t _btnTimer;
    byte _pin;
};
buttonMinim::buttonMinim(uint8_t pin) {
    pinMode(pin, INPUT_PULLUP);
    _pin = pin;
}
void buttonMinim::tick() {
    boolean btnState = digitalRead(_pin);
    if (!btnState && !flags.btnFlag && ((uint32_t)millis() - _btnTimer > 90)) {
        flags.btnFlag = true;
        _btnTimer = millis();
        flags.pressF = true;
        flags.holdedFlag = true;
    }
    if (btnState && flags.btnFlag && ((uint32_t)millis() - _btnTimer < 350)) {
        flags.btnFlag = false;
        _btnTimer = millis();
        flags.clickF = true;
        flags.holdF = false;
    }
    if (flags.btnFlag && ((uint32_t)millis() - _btnTimer > 900)) {
        if (!btnState) {
            flags.holdF = true;
        } else {
            flags.btnFlag = false;
            flags.holdF = false;
            _btnTimer = millis();
        }
    }
}
}

```

```

boolean buttonMinim::pressed() {
    buttonMinim::tick();
    if (flags.pressF) {
        flags.pressF = false;
        return true;
    }
    else return false;
}

boolean buttonMinim::clicked() {
    buttonMinim::tick();
    if (flags.clickF) {
        flags.clickF = false;
        return true;
    }
    else return false;
}

boolean buttonMinim::holding() {
    buttonMinim::tick();
    if (flags.holdF) {
        return true;
    }
    else return false;
}

boolean buttonMinim::holded() {
    buttonMinim::tick();
    if (flags.holdF && flags.holdedFlag) {
        flags.holdedFlag = false;
        return true;
    }
    else return false;
}

// различные функции
void serviceMode() {
    if (!digitalRead(BTN_PIN)) {
        byte serviceText[] = {_S, _E, _r, _U, _i, _C, _E};

```

```

disp.runningString(serviceText, sizeof(serviceText), 150);
lcd.setCursor(0, 0);
lcd.print("Режим: Сервис");
for (byte ii = 0; ii < 3; ii++)
{
    lcd.setCursor(0, 0);
    lcd.print("      ");
    delay(150);
    lcd.setCursor(0, 0);
    lcd.print("Режим: Сервис");
    delay(250);
}
while (!digitalRead(BTN_PIN)); // ждём отпущения
delay(200);
DEBUG("serviceMode");
lcd.setCursor(0, 0);
lcd.print("      ");
lcd.setCursor(0, 0);
lcd.print("Режим: Сервис");
long pumpTime = 0;
timerMinim timer100(100);
disp.displayInt(0);
bool flag;
for (;;) {
    enc.tick();
    if (timer100.isReady()) { // период 100 мс
        // работа помпы со счётчиком
        if (!digitalRead(ENC_SW)) {
            if (flag) pumpTime += 100;
            else pumpTime = 0;
            disp.displayInt(pumpTime);
            lcd.setCursor(0, 1);
            lcd.print("      ");
            lcd.setCursor(0, 1);
            lcd.print("Мсек: " + String(pumpTime));

```

```

        pumpON();
        flag = true;
    } else {
        pumpOFF();
        flag = false;
    }
}

// выводим объём и режим
void dispMode() {
    disp.displayInt(thisVolume);
    if (workMode)
    { disp.displayByte(0, _A);
      lcd.setCursor(0, 0);
      lcd.print("          ");
      lcd.setCursor(0, 0);
      lcd.print("Режим: Авто");
      lcd.setCursor(0, 1);
      lcd.print("          ");
      lcdupdate();
      DEBUG("Automatic Mode");
    }
    else {
        disp.displayByte(0, _P); DEBUG("Manual Mode");
        lcd.setCursor(0, 0);
        lcd.print("          ");
        lcd.setCursor(0, 0);
        lcd.print("Режим: Ручной");
        lcd.setCursor(0, 1);
        lcd.print("          ");
        lcdupdate();
    }
}

// наливайка, опрос кнопок
void flowTick() {
    if (FLOWdebounce.isReady()) {
        for (byte i = 0; i < NUM_SHOTS; i++) {

```

```

        if (!digitalRead(SW_pins[i]) && shotStates[i] == NO_GLASS) { // поставили
пустую рюмку
            timeoutReset(); // сброс таймаута
            shotStates[i] = EMPTY; // флаг на заправку
            LEDchanged = true;
            DEBUG("set glass № "+(String)i);
            //DEBUG(i);
        }
        if (digitalRead(SW_pins[i]) && shotStates[i] != NO_GLASS) { // убрали
пустую/полную рюмку
            shotStates[i] = NO_GLASS; // статус - нет рюмки
            LEDchanged = true;
            timeoutReset(); // сброс таймаута
            if ((systemState == PUMPING or systemState == MOVING) && i == curPumping)
{
                //секрет
                systemState = WAIT; // режим работы - ждать
                WAITtimer.reset();
                pumpOFF(); // ПОМПУ ВЫКЛ
                if (i == curPumping) curPumping = -1;
            }
            DEBUG("take glass № "+(String)i+" State: "+(String)systemState);
            //DEBUG(i);
        }
    }
    if (workMode) { // авто
        flowRoutnie(); // крутим обработку кнопок и поиск рюмок
    } else { // ручной
        if (btn.clicked()) { // клик!
            systemON = true; // система активирована
            timeoutReset(); // таймаут сброшен
        }
        if (systemON) flowRoutnie(); // если активны - ищем рюмки и всё такое
    }
}
}
}

```

```

// поиск и заливка
void flowRoutnie() {
if (btn.clicked()) { // клик!
    timeoutReset(); // таймаут сброшен
    }
if (systemState == SEARCH) { // если поиск рюмки
    bool noGlass = true;
for (byte i = 0; i < NUM_SHOTS; i++) {
    if (shotStates[i] == EMPTY && i != curPumping) { // поиск рюмки
        noGlass = false; // флаг что нашли хоть одну рюмку
        curPumping = i; // запоминаем выбор
        systemState = MOVING; // режим - движение
        shotStates[curPumping] = IN_PROCESS; // стакан в режиме заполнения
        DEBUG("find glass # "+(String)curPumping);
        //DEBUG(curPumping);
        break;
    }
}
if (noGlass) { // если не нашли ни одной рюмки
    systemON = false; // выключили систему
    DEBUG("no glass. State: "+(String)systemState);
    //DEBUG(systemState);
}
}
} else if (systemState == MOVING) { // движение к рюмке
    systemState = PUMPING; // режим - наливание
    FLOWtimer.setInterval(thisVolume * time50ml / 50); // перенастроили таймер
    FLOWtimer.reset(); // сброс таймера
    pumpON();
    lcd.setCursor(0, 1);
    lcd.print(" ");
    currentVolume = 0;
    FLOWtimer.timer = 0;
    DEBUG("Size: "+(String)thisVolume);
    //DEBUG(thisVolume);
}
}

```

```

    DEBUG("fill glass № "+(String)curPumping);
    //DEBUG(curPumping);
}
} else if (systemState == PUMPING) {           // если качаем
    currentVolume = (float)FLOWtimer.timer / time50ml * 50 + 1; //Наливаем
    disp.displayInt(int(currentVolume));
    DEBUG("Bulk-bulk: " + (String)int(currentVolume) + " gr.");
    lcd.setCursor(0, 1);
    lcd.print("V=" + (String)int(currentVolume) + "г");
    if (FLOWtimer.isReady()) {                 // если налили (таймер)
        pumpOFF();                             // помпа выкл
        totalGramm = totalGramm + thisVolume; //увеличиваем счетчик грамм
        writeValEeprom(readValEeprom(5)+thisVolume,5); //Записываем в EEPROM
грамм
        totalRumok++;                          //увеличиваем счетчик рюмок
        writeValEeprom(readValEeprom(3)+1,3); //Записываем в EEPROM рюмок
        dispMode();
        //lcdupdate();
        shotStates[curPumping] = READY;        // налитая рюмка, статус: готов
        curPumping = -1;                       // снимаем выбор рюмки
        systemState = WAIT;                    // режим работы - ждать
        //disp.displayInt(thisVolume);
        WAITtimer.reset();
        DEBUG("wait");
    }
} else if (systemState == WAIT) {
    if (WAITtimer.isReady()) {
        systemState = SEARCH;
        DEBUG("search");
    }
}
}
// отрисовка светодиодов по флагу (100мс)
void LEDtick() {
    if (LEDchanged && LEDtimer.isReady()) {

```

```

    LEDchanged = false;
}
}
// сброс таймаута
void timeoutReset() {
    if (!timeoutState) {
        disp.brightness(7);
        lcd.backlight();
    }
    timeoutState = true;
    TIMEOUTtimer.reset();
}
// сам таймаут
void timeoutTick() {
    if (timeoutState && TIMEOUTtimer.isReady()) {
        timeoutState = false;
        disp.brightness(1);
        lcd.noBacklight();
        POWEROFFtimer.reset();
    }

    // дёргаем питание серво, это приводит к скачку тока и powerbank не отключает
систему
    if (!timeoutState && TIMEOUTtimer.isReady()) {
        if (!POWEROFFtimer.isReady()) { // пока не сработал таймер полного
отключения
            } else {
                disp.clear();
            }
        }
    }
}
void lcdupdate() {
    //dispMode()
    lcd.setCursor(0, 1);
    lcd.print("      ");
}

```

```

    lcd.setCursor(0, 1);
    lcd.print("V=" + (String)int(thisVolume) + "r");
    lcd.setCursor(7, 1);
    lcd.print((String)totalRumok + "p/" + (String)totalGramm + "r");
}
void lcdinfo() {
    if (thisVolume >= 5 and posInfo == 0){
        dispMode();
        DEBUG("Volume: " + (String)thisVolume + " ml");
    }
    if (posInfo == 1) {lcd.setCursor(0, 1); lcd.print("Всего   ВКЛ:
" + (String)readValEeprom(1) + "   ");}
    if (posInfo == 2) {lcd.setCursor(0, 1); lcd.print("Всего   рЮМ:
" + (String)readValEeprom(3) + "   ");}
    if (posInfo == 3) {lcd.setCursor(0, 1); lcd.print("Всего   гр.:
" + (String)readValEeprom(5) + "   ");}
    if (posInfo == 4) {lcd.setCursor(0, 1); lcd.print("ИНФО 1   ");}
    if (posInfo == 5) {lcd.setCursor(0, 1); lcd.print("ИНФО 2   ");}
    if (posInfo == 6) {lcd.setCursor(0, 1); lcd.print("Версия: 211119   ");}
    DEBUG("posInfo: " + (String)posInfo);
}
byte val1 = EEPROM.read(addr);
byte val2 = EEPROM.read(addr+1);
int val = word(val1, val2);
return val;
}
byte hi = highByte(val); // старший байт
byte low = lowByte(val); // младший байт
EEPROM.write(addr, hi); // записываем в ячейку 1 старший байт
EEPROM.write(addr+1, low); // записываем в ячейку 2 младший байт
}
// кнопки-крутилки
enc.tick();
if (enc.isTurn()) {
    timeoutReset();
}

```

```

if (enc.isLeft()) {
  if (posInfo > 0)
  {
    posInfo -= 1;
    if (posInfo <= 0) {
      posInfo = 0;
    };
  }
  if (posInfo == 0) {
    thisVolume += 5;
  }
}
if (enc.isRight()) {
  thisVolume -= 5;
  if (thisVolume < 5)
  {
    thisVolume = 5;
    posInfo += 1;
    if (posInfo >= 6) {
      posInfo = 6;
    };
  }
}
dispMode();
lcdinfo();
writeValEeprom(thisVolume,7); //Записываем в EEPROM новый объем порции
}
}
if (btn.holded()) {
  timeoutReset();
  pumpOFF(); // ПОМПУ ВЫКЛ
  workMode = !workMode;
  posInfo = 0;
  dispMode();
}

```

```

if (encBtn.holded()) {
  lcd.backlight();
  lcd.setCursor(0, 1);
  lcd.print("Прокачка... ");
  DEBUG("keep");
  pumpON();
  while (!digitalRead(ENC_SW));
  timeoutReset();
  pumpOFF();
  delay(200);
  lcdupdate();
}
}
class encMinim
{
public:
  encMinim(uint8_t clk, uint8_t dt, uint8_t sw, boolean dir, boolean type);
  void tick();
  boolean isClick();
  boolean isTurn();
  boolean isRight();
  boolean isLeft();
  boolean isRightH();
  boolean isLeftH();
private:
  byte _clk, _dt, _sw;
  boolean _type = false;
  boolean _state, _lastState, _turnFlag, _swState, _swFlag, _turnState;
  byte _encState;
  uint32_t _debTimer;
  // 0 - ничего, 1 - лево, 2 - право, 3 - правоНажат, 4 - левоНажат
};

encMinim::encMinim(uint8_t clk, uint8_t dt, uint8_t sw, boolean dir, boolean type) {
  if (dir) {

```

```

    _clk = clk;
    _dt = dt;
} else {
    _clk = dt;
    _dt = clk;
}
_sw = sw;
_type = type;
pinMode (_clk, INPUT);
pinMode (_dt, INPUT);
pinMode (_sw, INPUT_PULLUP);
_lastState = digitalRead(_clk);
}
void encMinim::tick() {
    _encState = 0;
    _state = digitalRead(_clk);
    _swState = digitalRead(_sw);
    if (_state != _lastState) {
        _turnState = true;
        _turnFlag = !_turnFlag;
        if (_turnFlag || !_type) {
            if (digitalRead(_dt) != _lastState) {
                if (_swState) _encState = 1;
                else _encState = 3;
            } else {
                if (_swState) _encState = 2;
                else _encState = 4;
            }
        }
    }
    _lastState = _state;
}
if (!_swState && !_swFlag && millis() - _debTimer > 80) {
    _debTimer = millis();
    _swFlag = true;
    _turnState = false;
}

```

```

}
if (_swState && _swFlag && millis() - _debTimer > 80) {
    _debTimer = millis();
    _swFlag = false;
    if (!_turnState) _encState = 5;
}
}
boolean encMinim::isTurn() {
    if (_encState > 0 && _encState < 5) {
        return true;
    } else return false;
}
boolean encMinim::isRight() {
    if (_encState == 1) {
        _encState = 0;
        return true;
    } else return false;
}
boolean encMinim::isLeft() {
    if (_encState == 2) {
        _encState = 0;
        return true;
    } else return false;
}
boolean encMinim::isRightH() {
    if (_encState == 3) {
        _encState = 0;
        return true;
    } else return false;
}
boolean encMinim::isLeftH() {
    if (_encState == 4) {
        _encState = 0;
        return true;
    } else return false;
}

```

```

    }
    boolean encMinim::isClick() {
        if (_encState == 5) {
            _encState = 0;
            return true;
        } else return false;
    }
    // использован улучшенный алгоритм таймера на millis
    // алгоритм чуть медленнее, но обеспечивает кратные интервалы и защиту от
пропусков и переполнений
    class timerMinim
    {
    public:
        timerMinim(uint32_t interval);           // объявление таймера с
указанием интервала
        void setInterval(uint32_t interval);    // установка интервала работы таймера
        boolean isReady();                     // возвращает true, когда пришло время.
Сбрасывается в false сам (AUTO) или вручную (MANUAL)
        void reset();
        uint32_t timer;                        //
ручной сброс таймера на установленный интервал
    private:
        uint32_t _timer = 0;
        uint32_t _interval = 0;
    };
    timerMinim::timerMinim(uint32_t interval) {
        _interval = interval;
        _timer = millis();
    }
    void timerMinim::setInterval(uint32_t interval) {
        _interval = interval;
    }
    // алгоритм таймера v2.0
    boolean timerMinim::isReady() {
        uint32_t thisMls = millis();

```

```
timer = thisMls - _timer;
if (thisMls - _timer >= _interval) {
  do {
    _timer += _interval;
    if (_timer < _interval) break;    // переполнение uint32_t
  } while (_timer < thisMls - _interval); // защита от пропуска шага
  return true;
} else {
  return false;
}
}
void timerMinim::reset() {
  _timer = millis();
}
```