**ТОО «КАЗАХСКИЙ НИИ ВОДНОГО ХОЗЯЙСТВА»**

Отчет

О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

по мероприятию 3

«Разработка принципов и методов сбалансированного управления водораспределением на оросительных системах на основе гидрологической информации с учетом формирования водных ресурсов в бассейнах рек»

за период: 01.02-30.07.23 г.

**Исполнитель: Агарков Н.**

**Тараз 2023**

Оглавление

[ВВЕДЕНИЕ 3](#_Toc142384448)

[1. Разработка пользовательского интерфейса и структуры базы данных ИС 4](#_Toc142384449)

[1.1. Блок схемы расположения каналов 5](#_Toc142384450)

[1.2 Блок статуса и управления затворами 6](#_Toc142384451)

[1.3.Блок получения различных данных 8](#_Toc142384452)

[1.3.1 Раздел "Текущие показания датчиков на полях" 9](#_Toc142384453)

[1.3.2 Показания по гидропосту 10](#_Toc142384454)

[1.3.3 Расход воды по водоводу 11](#_Toc142384455)

[2. Взаимодействие ИС с базой данных. 12](#_Toc142384456)

[2.1 Подключение к базе данных. 13](#_Toc142384457)

[2.2 Анализ структуры базы данных 14](#_Toc142384458)

[3. Автоматический режим работы программы 17](#_Toc142384459)

[3.1 Автоматическое управление затворами по заранее определенному расписанию. 17](#_Toc142384460)

[3.2 Автоматическое управление затворами на основе показаний датчиков влажности. 18](#_Toc142384461)

[4. Определении ошибки при получении пакетов данных 21](#_Toc142384462)

[4.1 Мониторинг ошибок при работе сенсоров 21](#_Toc142384463)

[4.2 Отладка и исправление неполадок при работе программы. 24](#_Toc142384464)

[4.2.1 Ошибка при подключении к базе данных 24](#_Toc142384465)

[4.2.2 Зависание программы при обращении к записям базы данных 26](#_Toc142384466)

[4.2.3 Неправильная калибровка и настройка датчиков влажности 26](#_Toc142384467)

[5. Графическое представление взаимодействия датчиков влажности с программой 28](#_Toc142384468)

[ЗАКЛЮЧЕНИЕ 32](#_Toc142384469)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 33](#_Toc142384470)

# **ВВЕДЕНИЕ**

Управление системами орошения является важным и сложным процессом, который требует постоянного мониторинга и управления. Современные системы орошения обычно оснащены различными датчиками и устройствами, которые помогают операторам контролировать и управлять процессом орошения. Одним из таких устройств являются электрозатворы и датчики влажности и контроля микроклимата, которые используются для регулирования потока воды в системе.

Для эффективного управления системами орошения необходимо иметь программное обеспечение, которое бы могло контролировать и управлять работой электро-затворов. В данной работе рассматривается разработка программы для управления электро-затворами на пилотном участке с помощью языка программирования Pascal.

Основной задачей разработанной программы является обеспечение оператора системы орошения возможностью контролировать и управлять работой каждого затвора в системе, а также получать актуальную информацию от датчиков. Для этого в программе предусмотрены графический интерфейс и блок статуса и управления затворами. Графический интерфейс содержит схему расположения каналов, на которой отображаются местоположение всех затворов и датчиков на пилотном участке. Блок статуса и управления затворами предоставляет оператору системы возможность выбирать затвор, который нужно управлять, и задавать параметры его работы.

Программа реализована на языке программирования Pascal, который является объектно-ориентированным языком программирования и позволяет создавать качественные и эффективные программы. Для создания графического интерфейса использованы различные компоненты, такие как TButton, TLabel, TEdit, TListBox и др.

Разработанная программа позволяет операторам системы орошения контролировать и управлять работой электро-затворов на пилотном участке, что помогает повышать эффективность системы орошения в целом. Кроме того, графическая схема расположения каналов, включенная в программу, позволяет быстро и точно определить местоположение и состояние каждого элемента системы, что помогает операторам принимать более эффективные решения.

# **Разработка пользовательского интерфейса и структуры базы данных ИС**

**Схема разработанной программы**



Программа для управления электро-затворами на пилотном участке реализована на языке программирования Pascal, который является объектно-ориентированным языком программирования. Программа имеет пользовательский интерфейс, реализованный с помощью компонентов, таких как TButton, TLabel, TEdit, TListBox и др. Интерфейс предоставляет пользователю возможность выбрать затвор, который нужно управлять, и задать параметры его работы, а также подробную информацию о полученных данных с датчиков и расхода воды (Рисунок 1).

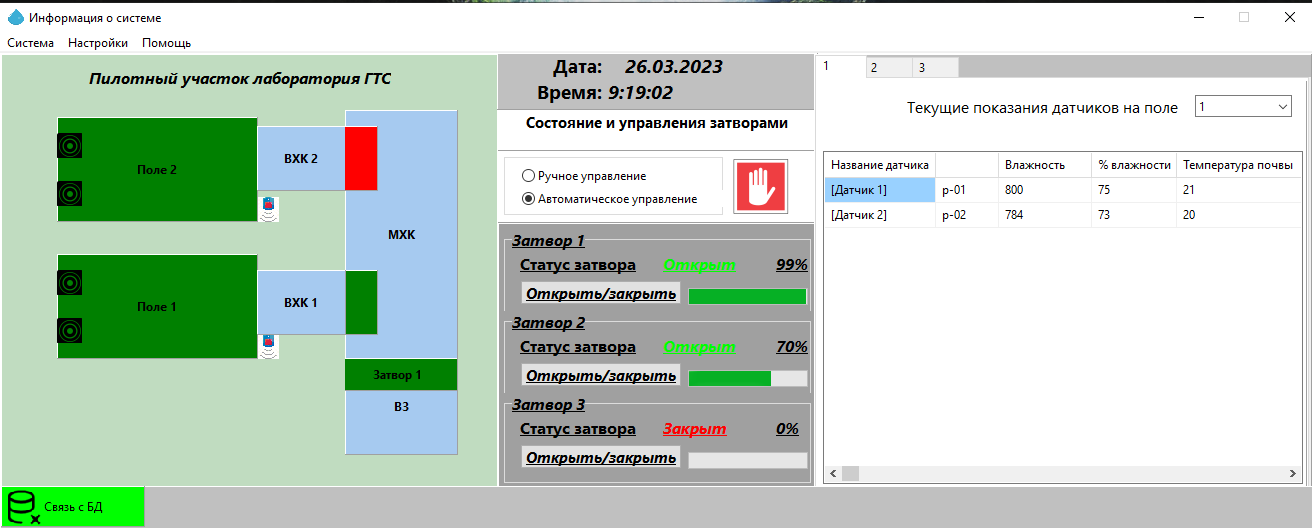


Рисунок 1. Главное окно системы

Сам интерфейс программы можно разбить на три основных блока:

## 1.1. Блок схемы расположения каналов

В этом блоке располагается схема расположения каналов, на которой отображается местоположение всех затворов и датчиков на пилотном участке. Это графическое изображение поля или схема системы орошения. С помощью данного блока можно отслеживать текущее состояние затворов. Для удобства пользователя цвета каждый затвор имеет независимый элемент, который окрашен в соответствующий цвет. (Рисунок 2)

На графической схеме расположения каналов отображены различные элементы, такие как затворы, регулирующие клапаны, гидропосты и датчики. Они помогают оператору системы быстро понять структуру системы и определить местонахождение и состояние каждого элемента.

Графическая схема расположения каналов может быть использована для различных целей, таких как планирование и моделирование системы, отладка и диагностика, контроль и управление. Она позволяет оператору системы быстро и точно определить местоположение и состояние каждого элемента системы, что помогает ему принимать более эффективные решения и сокращать время реакции на возможные проблемы [1].

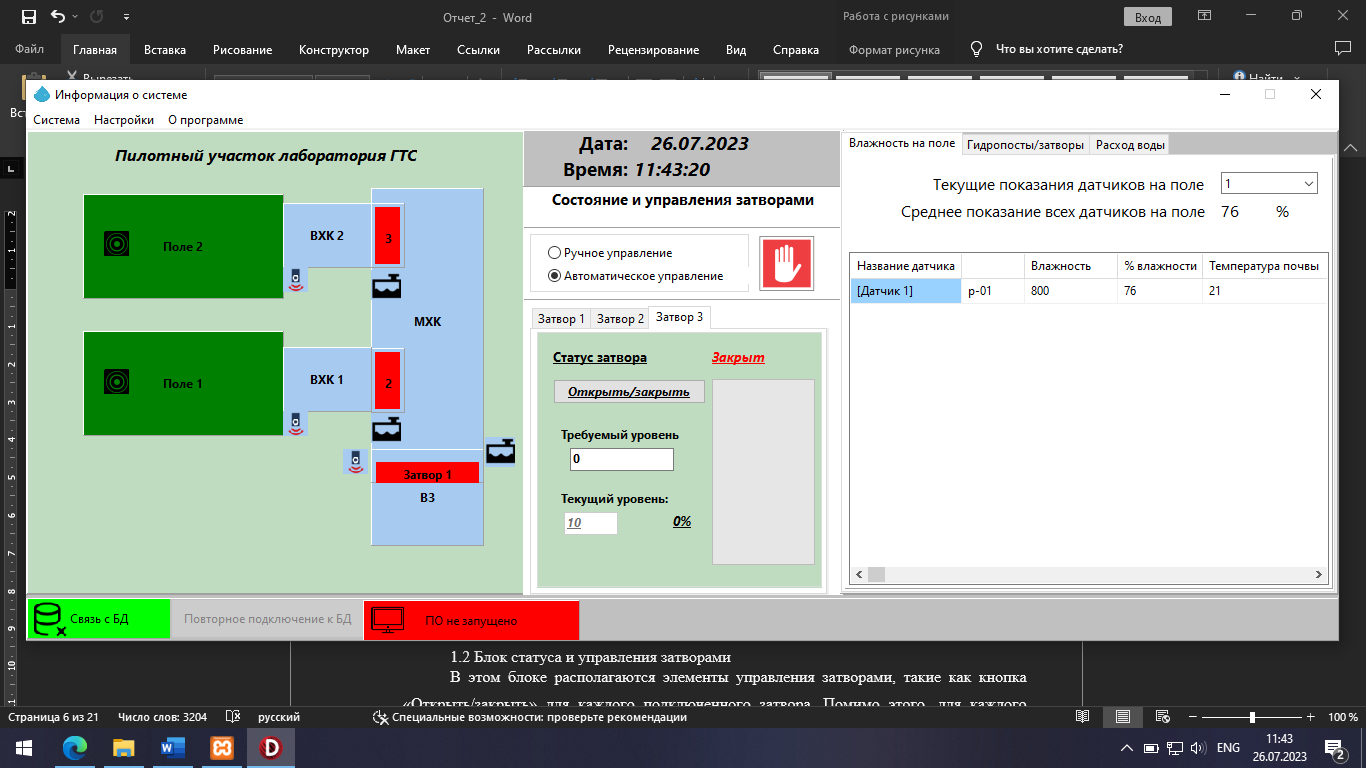


Рисунок 2. Графическая схема расположения каналов

## 1.2 Блок статуса и управления затворами

В этом блоке располагаются элементы управления затворами, такие как кнопка «Открыть/закрыть» для каждого подключенного затвора. Помимо этого, для каждого затвора показан процент его открытия, для более полного отображения всей ситуации. Эти данные могут обновляться в реальном времени в зависимости от состояния затворов и их положения. (рисунок 3.) Каждый затвор находится в отдельной вкладке

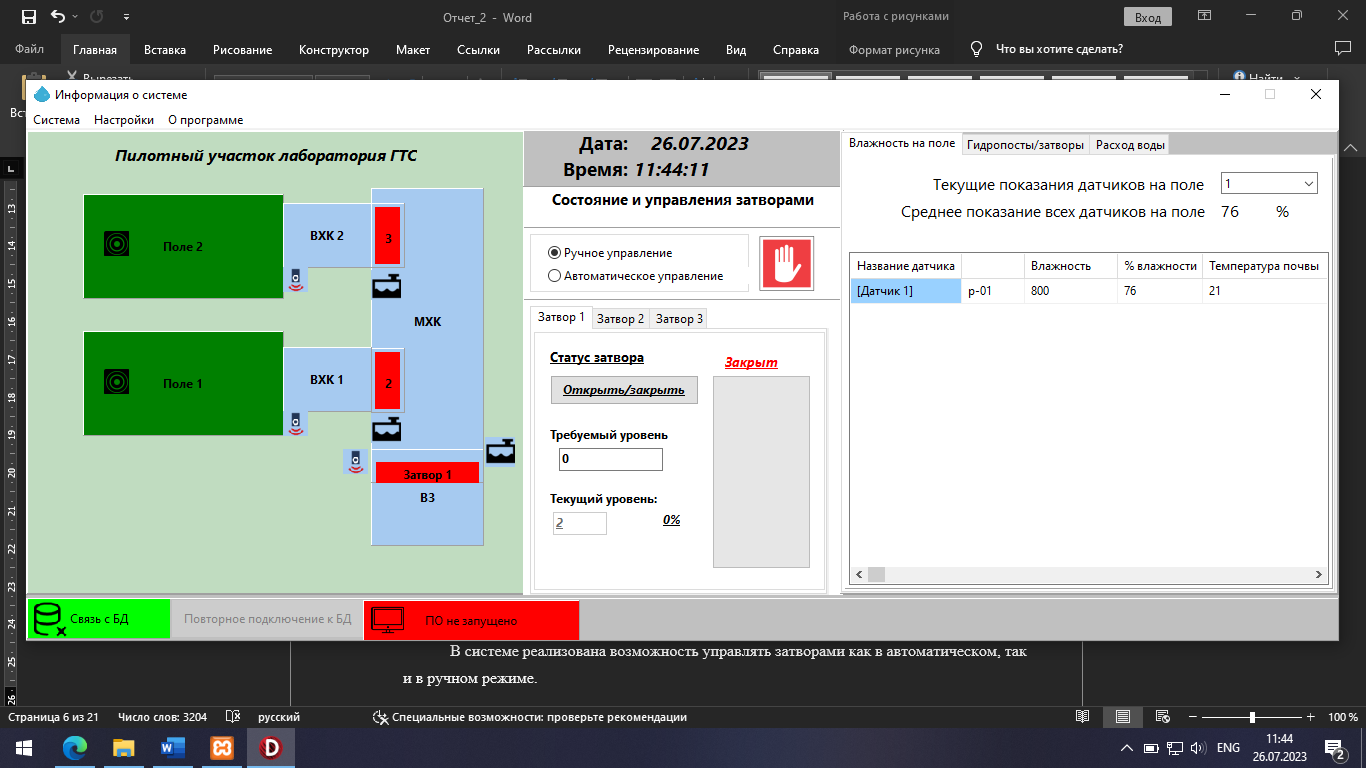


Рисунок 3. Блок состояний и управления затворами

В системе реализована возможность управлять затворами как в автоматическом, так и в ручном режиме.

В автоматическом режиме управление затворами осуществляется с помощью специального устройства (МСД), разработанного на основе модуля Arduino. Arduino - это платформа для создания электронных устройств, которая позволяет быстро и просто создавать прототипы и реализовывать различные проекты. Устройство, разработанное на базе Arduino, состоит из микроконтроллера, различных датчиков и модулей управления электро-затворами. В основе управления затворами лежит алгоритм, разработанный на языке программирования C++, который обеспечивает автоматическое открытие и закрытие затворов в зависимости от заданных параметров. Для взаимодействия с устройством, установленным на пилотном участке, используется специальное программное обеспечение, которое позволяет задавать параметры работы затворов, а также получать информацию о текущем состоянии системы. Все данные о состояниях затворов хранятся в базе данных.

При управлении системой в ручном режиме команды записываются в базу данных, из которой комплекс МСД берет данные. Использование базы данных для записи команд в ручном режиме является распространенным подходом в автоматизированных системах управления. Это позволяет сохранять и отслеживать все действия, совершаемые оператором системы, а также предоставлять доступ к этим данным другим пользователям. Кроме того, база данных может использоваться для автоматического выполнения задач, основанных на записанных командах.

Комплекс МСД может использовать данные из этой базы данных для мониторинга текущего состояния системы и выполнения соответствующих операций. Например, если оператор в ручном режиме отправил команду на открытие затвора, то комплекс МСД может обработать эту команду и открыть затвор в соответствии с заданными параметрами.

Кроме того, запись всех команд в базу данных может помочь при анализе работы системы, выявлении проблем и оптимизации процессов. Это может быть полезно при создании новых систем управления или при модернизации уже существующих.

При выборе пункта «Ручное управление» открывается возможность для отправки команд. При подаче команды, она записывается в БД, откуда МСД ее выполняет, по выполнению операции система обновляет статус затвора пример на рисунке 4.

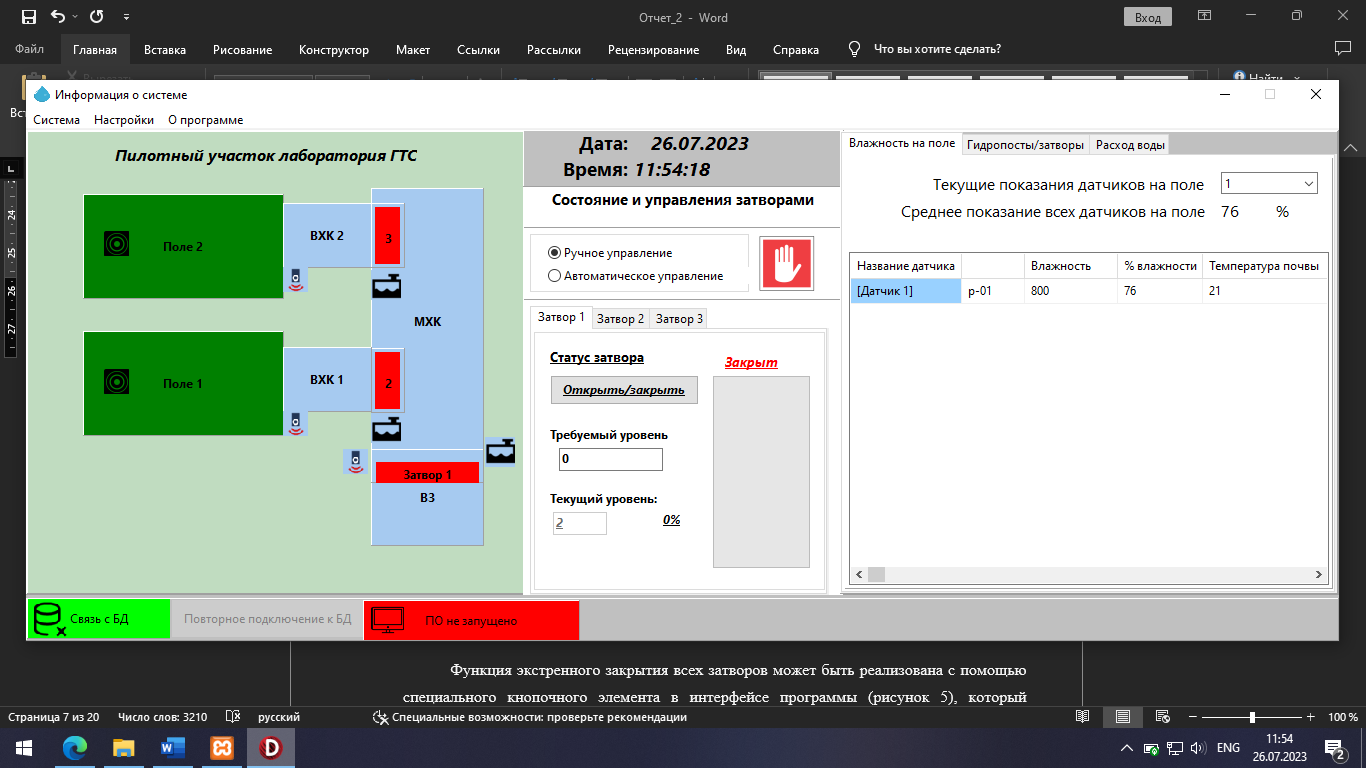


Рисунок 4. Ручное управление затворами

Так же в системе реализована функция экстренного закрытия всех затворов. Реализация функции экстренного закрытия всех затворов является важным элементом системы управления. Эта функция может быть использована в случае возникновения аварийной ситуации или при необходимости быстрого и надежного закрытия всех затворов.

При выполнении команды экстренного закрытия, комплекс МСД обрабатывает эту команду и отправляет соответствующие сигналы на управляющее устройство, которое в свою очередь закрывает все доступные затворы.

Функция экстренного закрытия всех затворов может быть реализована с помощью специального кнопочного элемента в интерфейсе программы (рисунок 5), который активирует эту функцию при нажатии.



Рисунок 5. Кнопка экстренного закрытия затворов

## 1.3.Блок получения различных данных

В этом блоке могут располагаться элементы для получения различных данных, таких как температура, влажность и расход воды. Эти данные могут быть взяты из датчиков, которые установлены на пилотном участке, и отображаться в режиме реального времени. Этот блок может помочь оператору системы мониторить текущие условия на пилотном участке и принимать решения на основе этих данных.

Сам блок для удобства пользователя был поделен на 3 раздела:

1.3.1 Раздел "Текущие показания датчиков на полях"

Раздел является важной частью интерфейса программы для удобства пользователя. Он предоставляет оператору системы информацию о текущем состоянии полей, а именно показания датчиков влажности и температуры почвы. В этом разделе пользователь может выбрать, какие поля ему необходимо отобразить - поле 1, поле 2 или все поля одновременно (рисунок 6, 7). Это позволяет оператору быстро получить информацию о состоянии конкретного поля или о состоянии всех полей сразу.

Информация о текущих показаниях датчиков может отображаться в виде таблицы. В таблице можно отображать показания датчиков в реальном времени, а также отслеживать изменения во времени. Также в системе предусмотрена оповещение о превышении определенных пороговых значений влажности или температуры почвы. Это позволяет оператору быстро отреагировать на возможные проблемы и принять соответствующие меры.

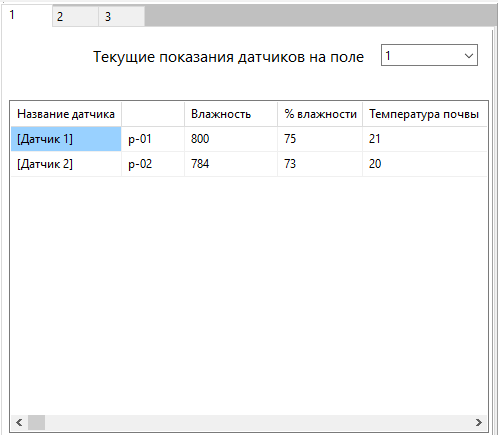


Рисунок 6. Текущие показания на поле 1.

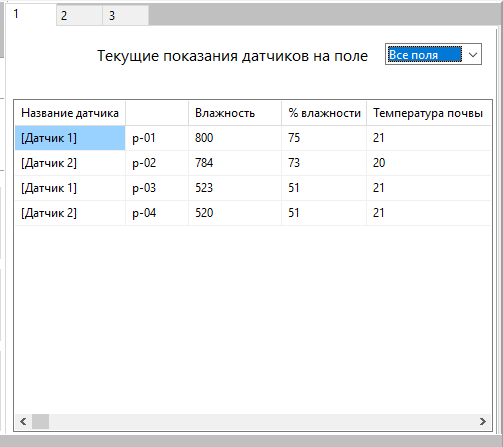


Рисунок 7. Текущие показания на всех полях.

1.3.2 Показания по гидропосту

Второй раздел отображает данные в реальном времени по гидропостам. Он включает информацию о состоянии затворов и показания тросиковых датчиков, а также уровне воды в канале в метрах. В этом разделе пользователь может просмотреть данные о состоянии каждого затвора, а также текущий уровень воды в канале. (Рисунок 8)

Для удобства пользователя информация может быть представлена в виде таблиц, позволяя быстро и легко получать необходимую информацию. Показания тросиковых датчиков позволяют определить уровень открытия затвора. Это может быть полезно для определения объема воды, пропущенной через канал, а также для определения текущей проходимости канала. Пользователь может установить заданный уровень открытия затвора, что позволяет оптимизировать проходимость канала и сохранять воду в необходимых объемах.

Этот раздел также может быть полезен для определения возможных проблем с затворами или с системой в целом. Если уровень воды в канале не соответствует заданному, или если тросиковые датчики показывают неправильные значения, оператор системы может быстро заметить проблему и принять меры для ее устранения.

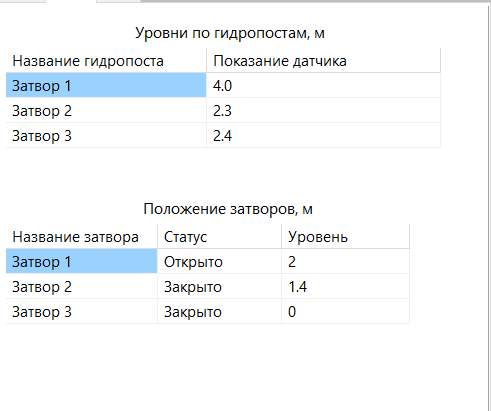


Рисунок 8. Данные датчиков по гидропостам

1.3.3 Расход воды по водоводу

Третий раздел интерфейса программы отображает данные о расходе воды по водоводам. В этом разделе пользователь может просмотреть данные о расходе воды на каждом водоводе, а также общий расход воды на всей системе.

Для удобства пользователю представлена таблица, позволяющая быстро оценить текущее состояние системы. Оператор системы может использовать эту информацию для оптимизации расхода воды, а также для выявления возможных проблем, таких как утечки или повреждения водоводов. Этот раздел может быть особенно полезен для оператора системы, который отвечает за контроль за расходом воды и за соблюдение установленных норм и правил использования водных ресурсов.[2] Он позволяет оператору быстро получить данные о текущем расходе воды и принимать необходимые меры в случае необходимости (рисунок 9).

Данные о расходе воды могут быть представлены в разных единицах измерения. В интерфейсе программы для удобства пользователя может быть выбрана единица измерения, в которой будут отображаться данные о расходе воды.

. В зависимости от требований и особенностей системы, может быть выбрана оптимальная единица измерения для отображения данных о расходе воды. Представление данных в различных единицах измерения может помочь пользователям быстро оценить текущее состояние системы и принимать необходимые меры для оптимизации использования водных ресурсов.

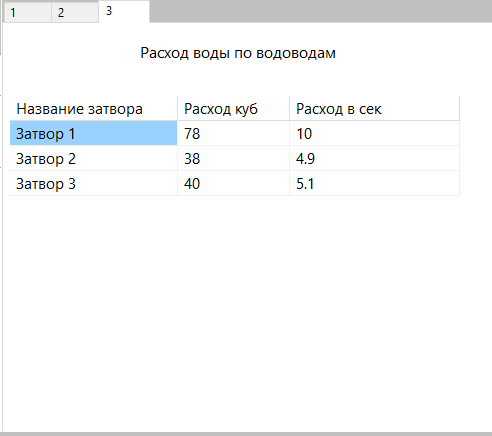


Рисунок 9. Показания расхода воды

# **Взаимодействие ИС с базой данных.**

Все данные о статусах системы, затворах, расходе воды и показаний датчиков находятся в базе данных MySQL.

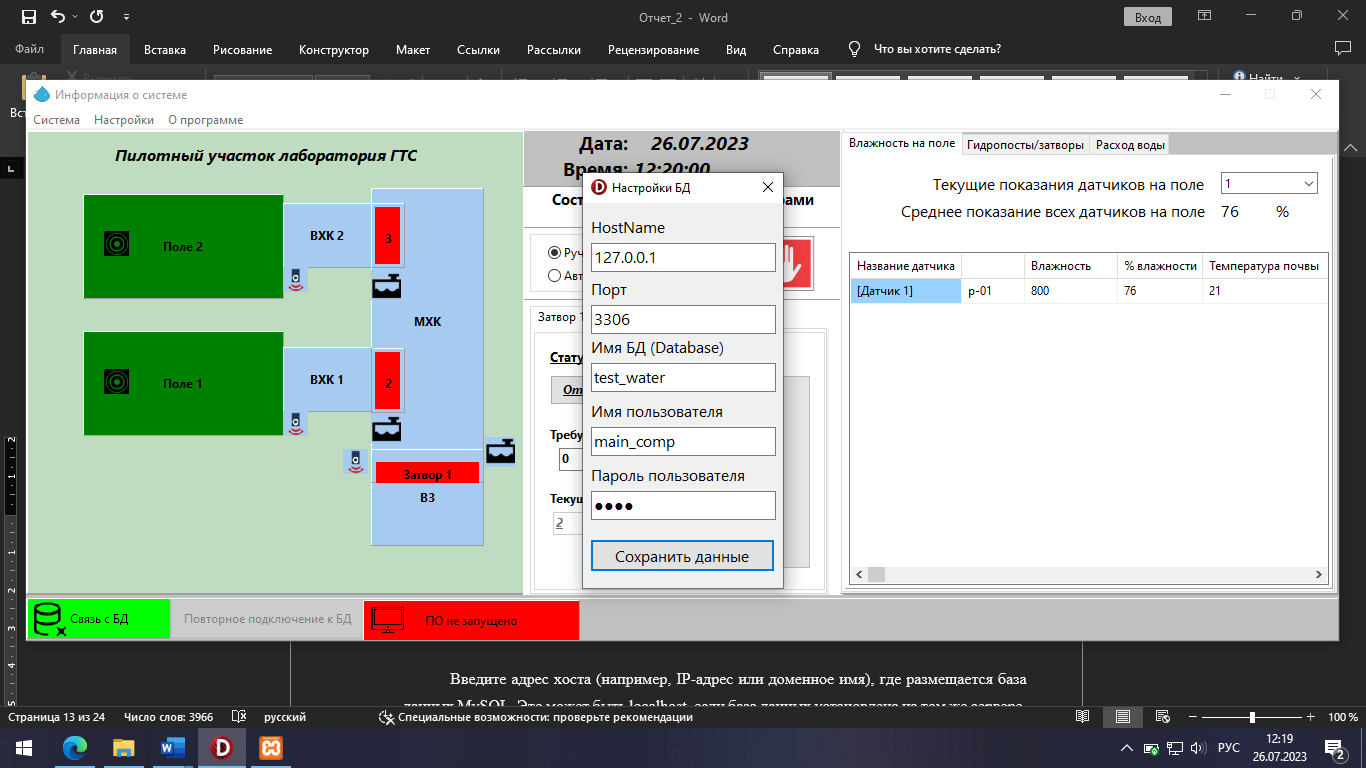
Использование базы данных MySQL для хранения данных является распространенной практикой в различных приложениях и системах. MySQL является реляционной базой данных, которая обеспечивает надежное и эффективное хранение данных. Она поддерживает множество функций и возможностей, таких как сжатие данных, резервное копирование, сетевое взаимодействие и многое другое.

Хранение данных в базе данных MySQL обеспечивает удобный доступ к данным, а также возможность быстрого поиска и обработки данных. База данных MySQL может использоваться для хранения большого объема данных, а также для поддержки множества пользователей и приложений. Для обеспечения безопасности и целостности данных в базе данных MySQL могут быть настроены соответствующие механизмы защиты данных, такие как шифрование, аутентификация и авторизация пользователей, а также механизмы контроля доступа. [3]

## 2.1 Подключение к базе данных.

Интеграция с СУБД MySQL предоставляет программе возможность эффективно хранить и управлять данными о гидрозатворах, заданиях по влажности и другой информацией, необходимой для автоматического управления системой орошения. Для обеспечения успешного подключения к базе данных MySQL и настройки соединения, предоставляем удобную форму, которая позволяет операторам указать актуальные данные для подключения.

Форма подключения к базе данных MySQL:



Хост (HostName): Введите адрес хоста (например, IP-адрес или доменное имя), где размещается база данных MySQL. Это может быть localhost, если база данных установлена на том же сервере, где выполняется программа.

Порт (Port): Укажите порт для подключения к базе данных MySQL. По умолчанию, порт MySQL - 3306, но если используется другой порт, убедитесь, что он указан правильно.

Имя пользователя (Username): Введите имя пользователя, которое будет использоваться для подключения к базе данных. Учетные данные пользователя должны иметь соответствующие права доступа для чтения и записи данных.

Пароль (Password): Введите пароль для указанного имени пользователя. Это обеспечивает безопасное подключение к базе данных и защиту от несанкционированного доступа.

База данных (Database): Выберите базу данных MySQL, в которой хранятся все необходимые таблицы для управления гидрозатворами и заданиями по влажности. Убедитесь, что база данных существует и имеет правильное название.

Проверить подключение:

Нажмите кнопку "Сохранить данные", чтобы проверить, успешно ли установлено соединение с базой данных. Если подключение прошло успешно, система сообщит об этом, и оператор сможет перейти к использованию программы. Если все данные для подключения корректны, программа автоматически сохраняет параметры подключения. Это обеспечивает автоматическое подключение при последующих запусках программы.

Форма для подключения к базе данных MySQL предоставляет операторам удобный способ настройки соединения и обеспечивает безопасное хранение и обработку данных. Это позволяет программе эффективно взаимодействовать с базой данных MySQL и обеспечивать надежное хранение и управление информацией для оптимального управления гидрозатворами и заданиями по влажности, а также анализом всех данных.

## 2.2 Анализ структуры базы данных

Краткий анализ структуры базы данных позволяет понять, какая информация хранится и как она организована для управления системой гидрозатворов и датчиков влажности. Вот основные выводы по каждой таблице:

*Таблица sensors\_hum (Общие сведения о датчиках):*

Эта таблица содержит общую информацию о датчиках влажности. Каждый датчик имеет уникальный идентификатор (Id\_sensors), который позволяет отслеживать его на поле. Здесь также указано идентификатор поля (Id\_pole), где расположен датчик. Поле Name содержит имя датчика, а Desck - краткое описание. Поле Active определяет, активен ли датчик.

*Таблица sensor\_level (Показание датчиков влажности):*

Эта таблица связана с таблицей sensors\_hum и содержит показания датчиков влажности. Каждая запись в таблице содержит идентификатор датчика (Id\_sensors), а также неизменяемое значение влажности (Hum\_val) и значение влажности в процентах (Hum\_val\_proc), полученное с помощью функции map в Arduino. Поле Temp\_val содержит показание температуры.

*Таблица status\_zatvor (Уровень затворов и их статус):*

Эта таблица содержит информацию о статусе и уровне открытия затворов. Каждый затвор имеет отображаемое имя (Name) и статус (Status), где 1 - открыт, 0 - закрыт. Поле Level\_metr содержит уровень открытия шлюза, а Procent\_otk - процент открытия. Open\_level хранит необходимый уровень открытия затвора, а Id\_pole связывает затвор с соответствующим полем. average\_value\_h содержит среднее арифметическое показаний датчиков влажности в процентах на соответствующем поле.

*Таблица task\_humidity:*

В этой таблице хранятся задания по влажности для управления затворами. Поле Id\_zatvor связывает задание с конкретным затвором. Поле Min определяет минимальный уровень поддержания влажности, а Max - максимальный уровень поддержания влажности.

*Таблица task\_schedule:*

Эта таблица содержит задания по расписанию для управления затворами. Id\_zatvor связывает задание с определенным затвором. Time\_open и Time\_close определяют время открытия и закрытия затвора.

*Таблица error:*

В этой таблице хранятся недавние ошибки, возникшие в системе. Поле Item указывает на элемент, вызвавший ошибку (например, затвор или датчик). Code\_error указывает на наличие ошибки (0 - нет ошибок, 1 - есть ошибка), а Text\_error содержит текстовое описание ошибки. Value\_programm используется для определения запущенной второй программы.

Таблица sensors, которая хранит архив данных по уровням затвора, влажности и другим параметрам, представляет ценный инструмент для анализа и отслеживания исторических данных системы орошения. Ее структура позволяет хранить и организовывать записи об изменениях в различных параметрах со временем. Вот краткое описание этой таблицы:

*Таблица sensors (Архив данных по уровням затвора, влажности и т.д.):*

id (int(11)): Уникальный идентификатор записи, автоматически увеличивающийся при добавлении новых данных. Это основной ключ таблицы.

sensor\_id (int(11)): Идентификатор датчика, к которому относится запись. Указывает на связанный датчик, чтобы определить, от какого именно датчика получены данные.

level (int(11)): Значение уровня, полученное от датчика. Представляет числовое значение уровня.

bat (int(11)): Значение заряда батареи, связанное с датчиком. Представляет числовое значение уровня заряда.

rashod (int(11)): Значение расхода, связанное с датчиком. Представляет числовое значение расхода.

date\_insert (datetime): Временная метка, указывающая на момент вставки записи в таблицу. Отражает дату и время добавления данных.

reset (varchar(10)): Текстовое поле, содержащее информацию о сбросе. Вероятно, хранит информацию о сбросе данных датчика.

lastcode (int(11)): Таким образом, эта таблица records содержит записи о различных параметрах, связанных с датчиками, и их соответствующими значениями, дату и время вставки записи, а также информацию о сбросе и статусе. Эти данные могут быть использованы для анализа, мониторинга и оптимизации процессов орошения и системы в целом.

Таблица sensors предоставляет ценные данные для анализа и мониторинга прошлых событий и изменений в системе орошения. Поиск и анализ архивных данных позволяет выявлять тенденции, прогнозировать потребности в водоснабжении и оптимизировать работу системы, основываясь на предыдущих измерениях и результаты.

Анализ структуры базы данных позволяет сделать вывод, что она хорошо организована для хранения и управления данными, необходимыми для автоматического управления гидрозатворами и датчиками влажности на основе полученных показаний. Связи между таблицами позволяют эффективно использовать данные и обеспечивать оптимальное функционирование системы орошения.

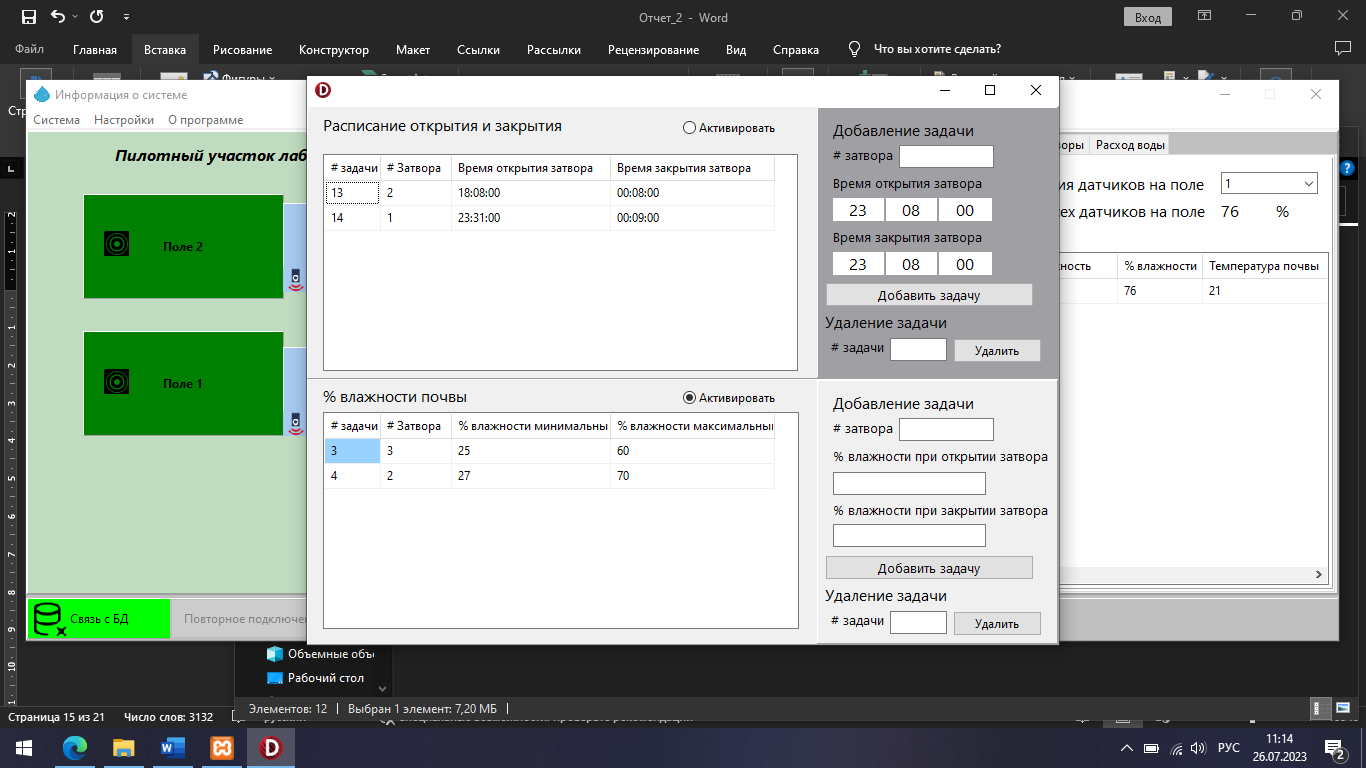
# **Автоматический режим работы программы**

В автоматическом режиме, программа предлагает удобное расписание управления гидрозатворами по времени, позволяя задавать точные интервалы для открытия и закрытия затворов, а также управлении по внешним сенсорам датчиков влажности. Этот режим позволяет автоматически адаптироваться к меняющимся потребностям полевых условий, оптимизируя использование ресурсов воды и обеспечивая максимальную эффективность орошения. При запуске программы система автоматически запускается в автоматическом режиме управления затворами.

## 3.1 Автоматическое управление затворами по заранее определенному расписанию.

Для создания задания в расписании, позволяющего автоматически управлять гидрозатворами на основе времени, требуется предоставить оператору интерфейс, который позволит установить необходимые параметры.

Форма создания задания в расписании:



В этой форме вы можете указать параметры для каждого задания, которые определяют, когда и на какой затвор применить автоматическое управление.

Параметры задания:

* Выберите затвор: Укажите номер к которому применяется данное задание.
* Время открытия: Задайте точное время, когда выбранный затвор должен быть открыт.
* Время закрытия: Задайте точное время, когда выбранный затвор должен быть закрыт.
* Активность: Определите, должно ли задание быть активным или выключенным. Выключенные задания не будут выполняться в установленное время.

Вы можете создавать несколько заданий для разных затворов и временных интервалов. Система будет автоматически выполнять задания согласно указанным параметрам.

Интерфейс для управления заданиями: После создания задания, вы можете просматривать, и удалять их в будущем через интерфейс управления заданиями. Это позволяет вам легко вносить изменения в расписание, а также отключать и включать задания по мере необходимости.

Важно:

Убедитесь, что указанные временные интервалы для заданий не перекрываются, а также имеется только одно задание для каждого затвора, чтобы избежать конфликтов при управлении затворами. Система будет выполнять задания строго согласно указанным временам, поэтому необходимо внимательно указывать параметры.

При удалении задачи в расписании необходимо указать идентификатор (ID) задания, чтобы определить точно, какое задание следует удалить из расписания. Введение этого параметра предотвращает случайное удаление или нежелательное воздействие на другие задания.

## 3.2 Автоматическое управление затворами на основе показаний датчиков влажности.

Автоматическое управление затворами на основе показаний датчиков влажности представляет передовую технологию, которая позволяет оптимизировать процесс орошения полей, поддерживая оптимальные уровни влажности для растений. Наш алгоритм для автоматического управления затворами на основе данных датчиков влажности обеспечивает эффективное использование водных ресурсов и повышение урожайности.

Алгоритм инициализации: При запуске программы, алгоритм проверяет таблицу task\_humidity и инициализирует параметры для каждого затвора, такие как минимальный уровень влажности (Min) и максимальный уровень влажности (Max), которые должны поддерживаться на каждом поле.

Проверка данных датчиков: Алгоритм периодически опрашивает данные датчиков влажности на каждом поле и получает текущее значение влажности для каждого затвора. Анализ текущей влажности: На основе полученных данных датчиков, алгоритм сравнивает текущий уровень влажности с заданными значениями минимальной (Min) и максимальной (Max) влажности из таблицы task\_humidity.

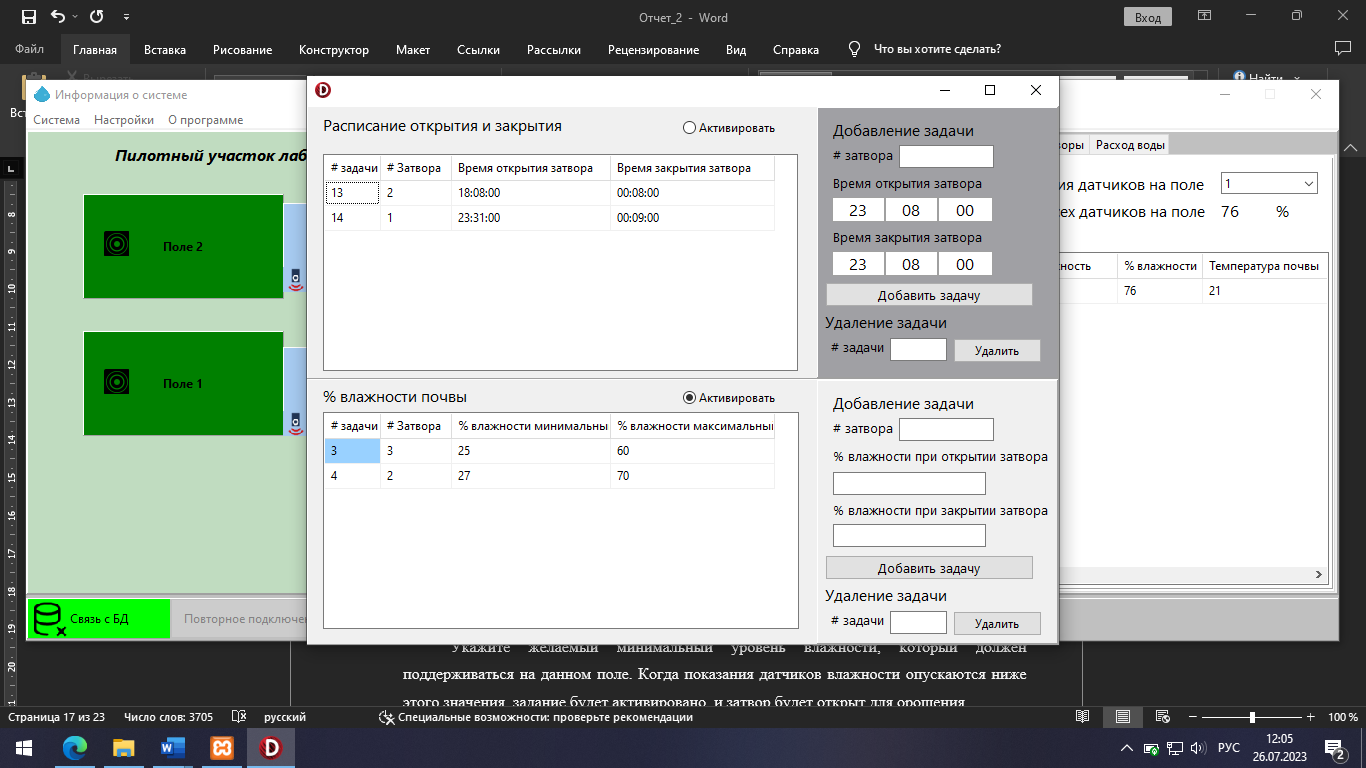
Принятие решения об открытии или закрытии затвора: Если текущий уровень влажности находится ниже минимального значения (Min), алгоритм принимает решение открыть затвор, чтобы обеспечить необходимое орошение и поддерживать оптимальный уровень влажности.

Продолжение проверки: После открытия затвора, алгоритм продолжает проверять данные датчиков и текущий уровень влажности. Если текущая влажность становится выше максимального значения (Max), алгоритм принимает решение закрыть затвор.

Оптимизация орошения: Алгоритм динамически регулирует время работы затворов, чтобы обеспечить равномерное и оптимальное орошение на каждом поле, основываясь на показаниях датчиков влажности.

Форма добавления задания по влажности предоставляет операторам удобный способ настройки автоматического управления затворами на основе показаний датчиков влажности. Ниже приведена детальная описание формы:

Форма добавления задания по влажности:



Выбор затвора:

Выберите затвор, к которому хотите добавить задание. Каждый затвор имеет уникальный идентификатор (Id\_zatvor), что позволяет точно указать, к какому затвору применяется задание.

Минимальный уровень влажности (Min): Укажите желаемый минимальный уровень влажности, который должен поддерживаться на данном поле. Когда показания датчиков влажности опускаются ниже этого значения, задание будет активировано, и затвор будет открыт для орошения.

Максимальный уровень влажности (Max): Введите максимальный уровень влажности, который следует поддерживать на данном поле. Когда показания датчиков влажности достигают или превышают этот уровень, задание активируется, и затвор будет закрыт для экономии водных ресурсов.

Активность задания: Установите параметр активности задания. Если задание активно, то оно будет выполняться согласно заданным условиям. Если вы хотите временно выключить задание, просто снимите флажок активности.

Сохранить задание: После того, как вы заполнили все необходимые параметры, нажмите кнопку "Добавить задачу", чтобы добавить задание в расписание управления затворами на основе влажности.

Удаление задачи: Если вы хотите удалить задание, укажите ID задачи и нажмите кнопку "Удалить". Данный процесс невозможно остановить.

Просмотр заданий: после сохранения задания, вы можете просматривать его параметры в будущем через интерфейс управления заданиями. Это позволяет вам удалять или временно отключать их по мере необходимости.

Форма добавления задания по влажности обеспечивает простой и интуитивно понятный способ настройки автоматического управления затворами, позволяя точно контролировать уровни влажности на каждом поле и оптимизировать процессы орошения для максимальной эффективности и урожайности.

# **Определении ошибки при получении пакетов данных**

## 4.1 Мониторинг ошибок при работе сенсоров

Программа также содержит алгоритмы для обработки ошибок и исключительных ситуаций, таких как потеря связи с контроллером затвора или неправильный ввод параметров. В целом, программа для управления электро-затворами на пилотном участке, представляет собой надежное и удобное решение для автоматизации процесса управления затворами.

Обработка ошибок

Таблица ошибок: Описание структуры таблицы ошибок в базе данных и её роль в записи возникающих проблем.

Ниже представлено более подробное описание структуры таблицы ошибок:

*Таблица "error":*

Item: Поле, которое указывает на элемент системы, вызвавший ошибку. Это может быть имя датчика или затвора, который был замечен в некорректном состоянии. Например: "Gate1" для затвора 1, "Sensor\_gate\_level\_2" для датчика уровня на затворе 2 и т.д.

Code\_error: Это поле хранит информацию о наличии ошибки. Значение "0" означает, что ошибок не обнаружено, а значение "1" указывает на наличие ошибки. Это поле может использоваться для быстрой проверки наличия ошибок без необходимости считывать текстовое описание ошибки.

Text\_error: Поле, которое содержит текстовое описание ошибки, возникшей у соответствующего элемента (указанного в поле Item). Например, текст ошибки может быть "Низкий уровень воды" или "Сбой датчика влажности".

Value\_programm: Это поле используется для определения, запущена ли вторая программа или нет. Значение может быть инкрементировано второй программой, чтобы сообщить об этом факте основной программе или использоваться для синхронизации между различными частями программы.

|  |  |  |
| --- | --- | --- |
| № | Параметр (Item) | Описание параметра |
|  | Gate1 | Затвор 1 |
|  | Gate2 | Затвор 2 |
|  | Gate3 | Затвор 3 |
|  | Sensor\_gate\_level\_1 | Тросиковый датчик на затворе 1 |
|  | Sensor\_gate\_level\_2 | Тросиковый датчик на затворе 2 |
|  | Sensor\_gate\_level\_3 | Тросиковый датчик на затворе 3 |
|  | Senor\_water\_level\_1 | Датчик уровня воды 1 |
|  | Senor\_water\_level\_2 | Датчик уровня воды 2 |
|  | Senor\_water\_level\_3 | Датчик уровня воды 3 |
|  | Sensor\_humidity\_1 | Датчик влажности 1 |
|  | Sensor\_humidity\_2 | Датчик влажности 2 |

Роль таблицы ошибок:

* Запись ошибок: Когда программа обнаруживает некорректные значения датчиков или проблемы с работой затворов, она записывает соответствующую информацию в таблицу ошибок. Это позволяет узнать, какие именно элементы вызывают проблемы и какие ошибки произошли.
* Отображение ошибок: Пользовательский интерфейс программы может обращаться к таблице ошибок для отображения текстового описания ошибок, что поможет оператору системы быстро выявить и понять проблемы.
* Отслеживание состояния системы: При использовании таблицы ошибок программа может периодически проверять её содержимое для выявления новых ошибок. Это позволяет оперативно реагировать на возможные неполадки и предотвращать нежелательные последствия.

Оповещение оператора о возникновении ошибки путем мигания элемента на схеме и прекращение мигания после нормализации работы может быть реализовано с помощью анимации и обработки состояния элементов. Вот подробное описание этого процесса:

*Обнаружение ошибки:*

Когда программа обнаруживает ошибку у датчика или затвора, она записывает соответствующую информацию в таблицу ошибок, как уже было описано ранее.

После обнаружения ошибки, программа также изменяет состояние соответствующего элемента на схеме, делая его способным к анимации (например, путем изменения его цвета на более яркий или мигания).

*Мигание элемента на схеме:*

Когда состояние элемента изменено на "ошибка", он начинает мигать. Мигание может быть реализовано путем переключения его видимости (отображения и скрытия) или изменения цвета на протяжении заданных интервалов времени.

Например, элемент может быть видимым в течение 0.5 секунды, затем скрываться на 0.5 секунды и так далее, создавая впечатление мигания.

*Прекращение мигания после нормализации работы:*

Когда программа исправляет ошибку и система возвращается в нормальное состояние, она обновляет информацию в таблице ошибок, устанавливая значение Code\_error на "0", чтобы указать, что ошибок больше нет.

Кроме того, программа изменяет состояние элемента на схеме обратно на "нормальное", тем самым прекращая мигание.

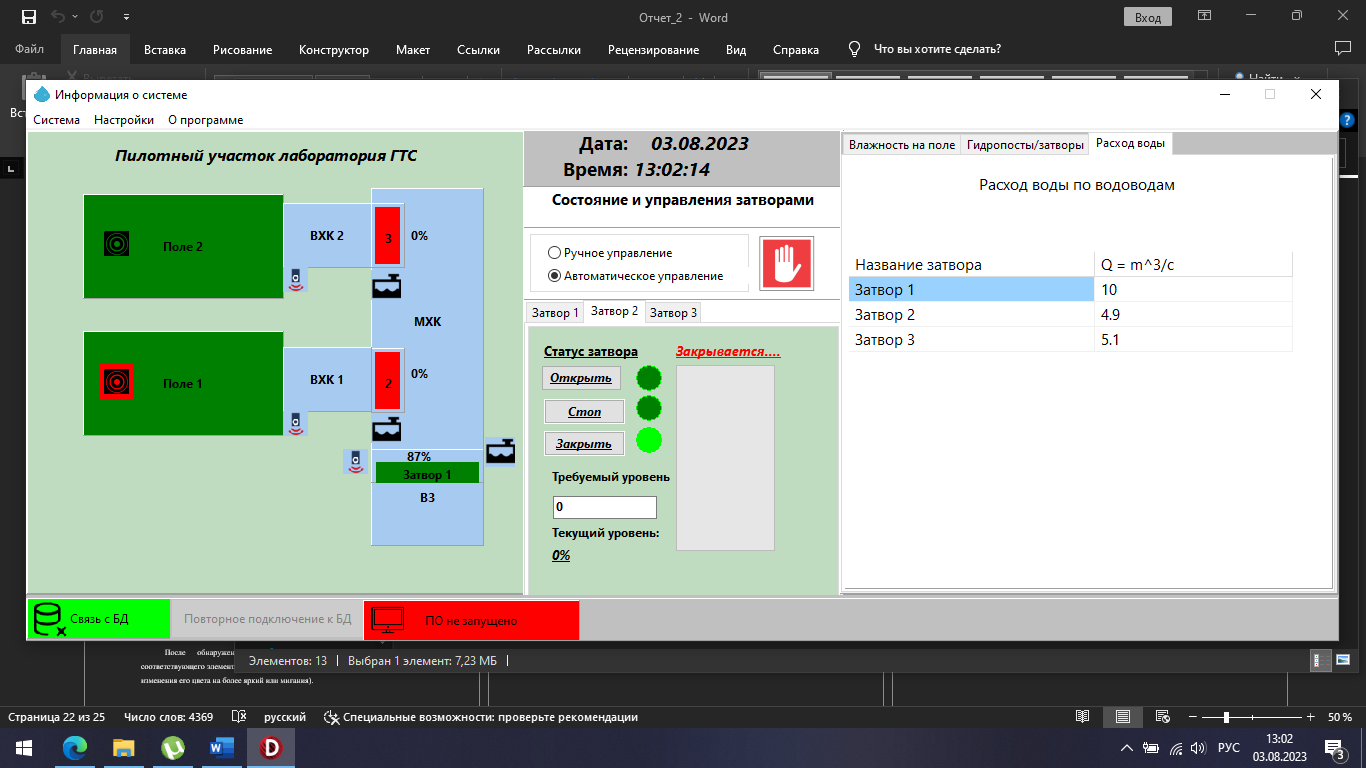
*Обновление данных в реальном времени:*

Важно, чтобы визуальное оповещение и состояние элементов на схеме отражали текущее состояние системы в реальном времени. Поэтому программа должна периодически проверять таблицу ошибок и обновлять состояние элементов на основе актуальной информации.

Таким образом, процесс визуального оповещения оператора будет следующим:

* При возникновении ошибки, элемент на схеме, соответствующий ошибочному датчику или затвору, начинает мигать.
* Когда ошибка устраняется и система возвращается в нормальное состояние, элемент прекращает мигание и возвращается в обычное состояние.

Пример работы системы с ошибкой показан на рисунке ниже



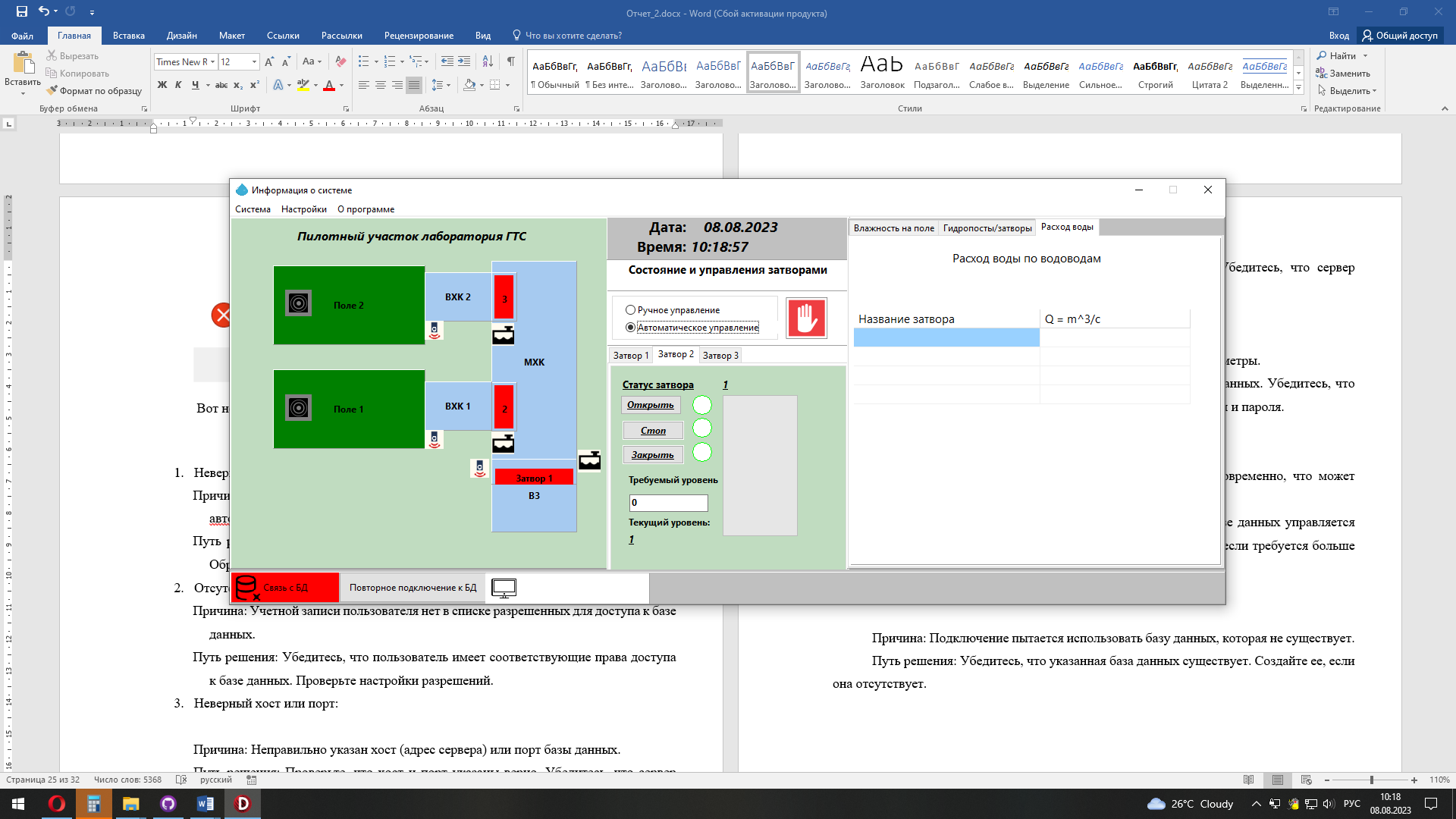
## **4.2 Отладка и исправление неполадок при работе программы.**

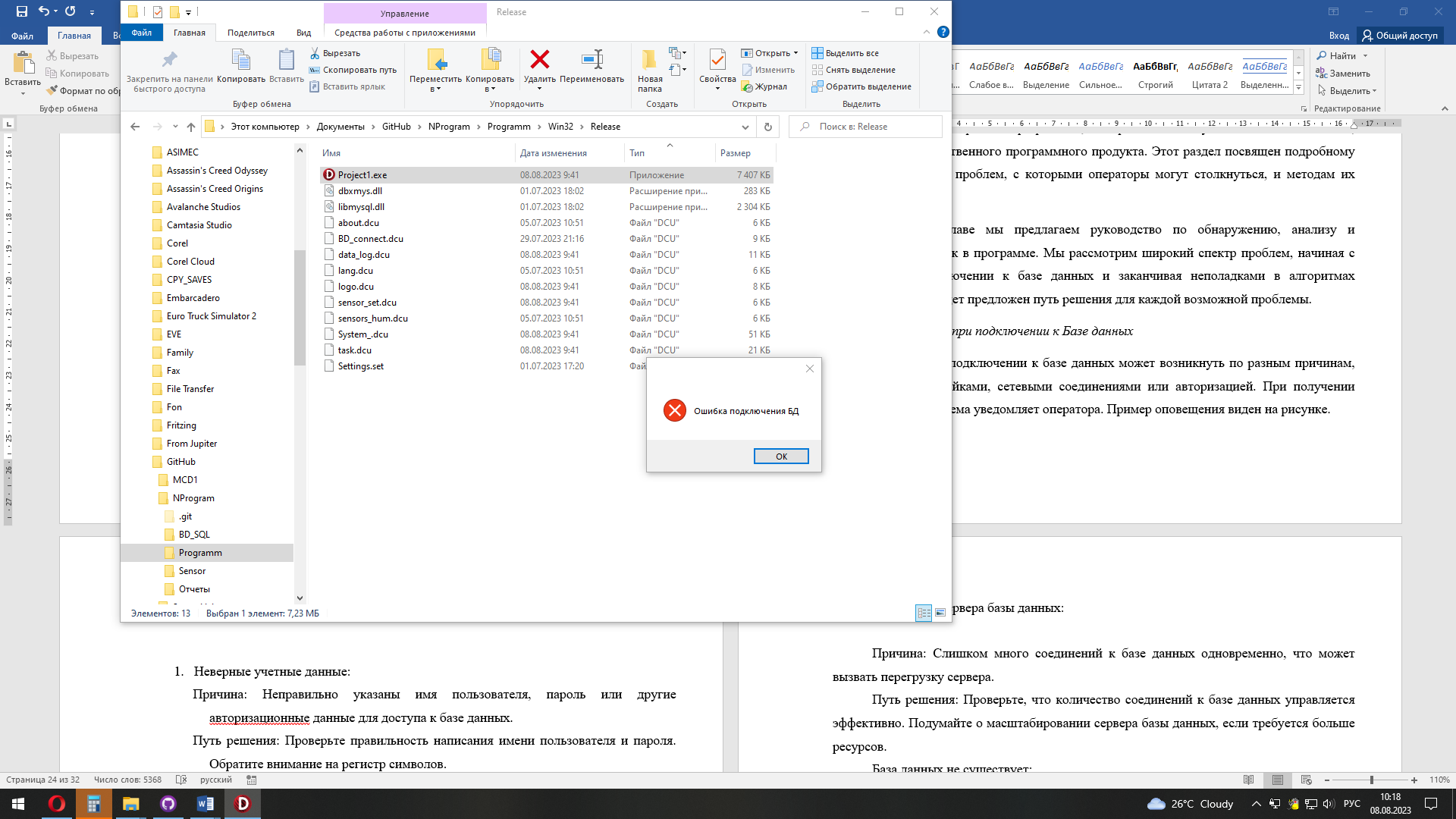
При разработке программного обеспечения невозможно избежать возникновения ошибок и неполадок. Эти ситуации, скорее всего, возникнут в процессе тестирования программы, когда она вступает в контакт с реальными данными и сценариями использования. Однако важно понимать, что отладка и устранение ошибок — это неотъемлемая часть процесса разработки, которая способствует созданию надежного, стабильного и качественного программного продукта. Этот раздел посвящен подробному анализу возможных проблем, с которыми операторы могут столкнуться, и методам их устранения.

В данной главе мы предлагаем руководство по обнаружению, анализу и исправлению ошибок в программе. Мы рассмотрим широкий спектр проблем, начиная с ошибок при подключении к базе данных и заканчивая неполадками в алгоритмах управления. Вам будет предложен путь решения для каждой возможной проблемы.

***4.2.1 Ошибка при подключении к базе данных***

Ошибка при подключении к базе данных может возникнуть по разным причинам, связанным с настройками, сетевыми соединениями или авторизацией. При получении данной ошибки система уведомляет оператора. Пример оповещения виден на рисунке.



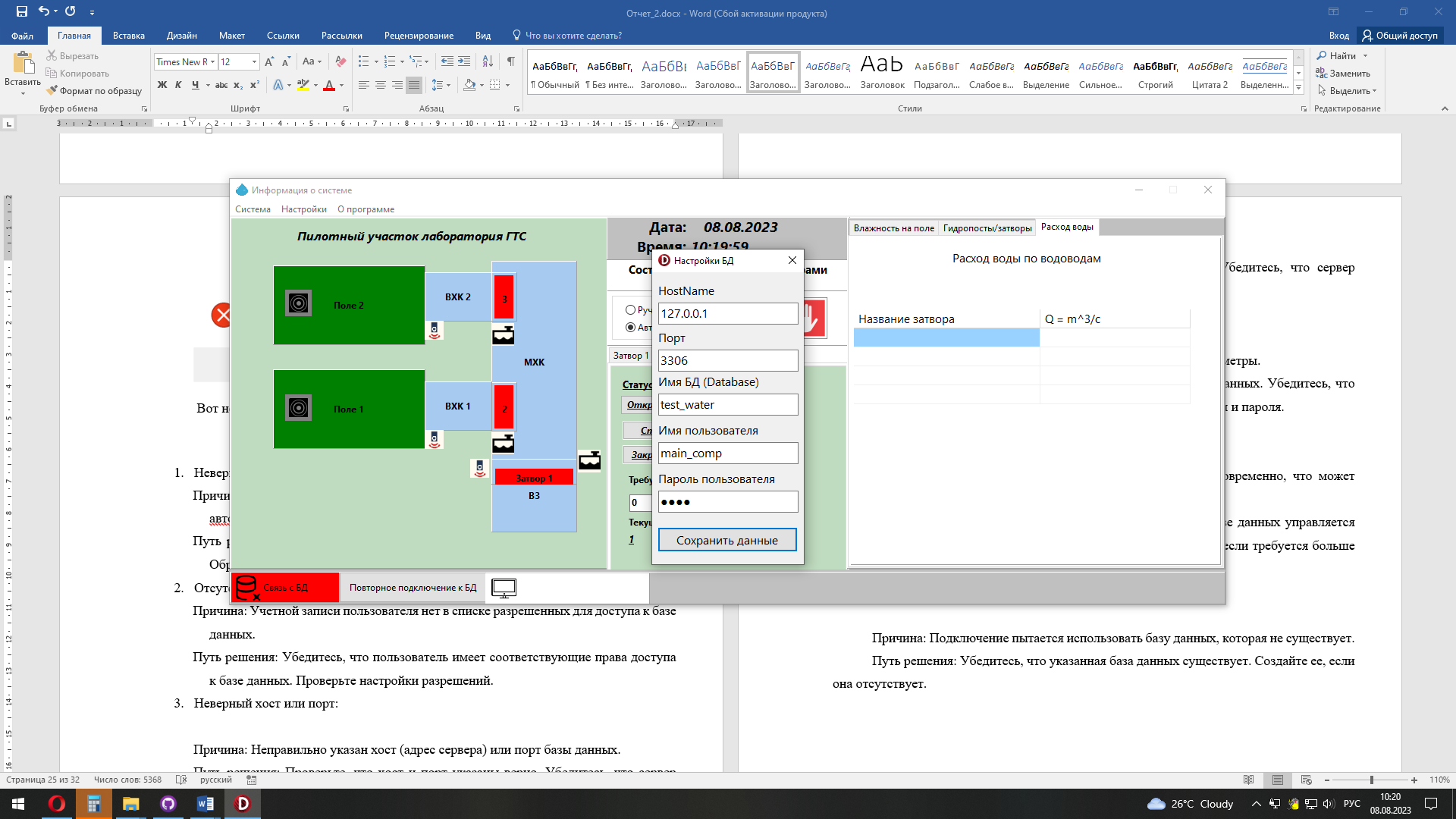


Вот некоторые из возможных причин и пути их решения:

1. **Неверные учетные данные:**

Причина: Неправильно указаны имя пользователя, пароль или другие авторизационные данные для доступа к базе данных.

Путь решения: Проверьте правильность написания имени пользователя и пароля. Обратите внимание на регистр символов. При необходимости исправьте на актуальные значения в следующей вкладке. Настройки –>Подключение к БД



1. **Отсутствие доступа:**

Причина: Учетной записи пользователя нет в списке разрешенных для доступа к базе данных.

Путь решения: Убедитесь, что пользователь имеет соответствующие права доступа к базе данных. Проверьте настройки разрешений или верные данные пользователя.

1. **Неверный хост или порт:**

Причина: Неправильно указан хост (адрес сервера) или порт базы данных.

Путь решения: Проверьте, что хост и порт указаны верно. Убедитесь, что сервер базы данных доступен по указанному адресу. При необходимости исправьте на актуальные значения в следующей вкладке. Настройки –>Подключение к БД

1. **Проблемы с сетью:**

Причина: Проблемы с сетевым соединением между приложением и сервером базы данных.

Путь решения: Проверьте сетевое соединение между компьютером, на котором работает приложение, и сервером базы данных. Убедитесь, что нет блокирующих брандмауэров или прокси-серверов.

1. **Сервер базы данных недоступен:**

Причина: Сервер базы данных выключен, недоступен или не функционирует.

Путь решения: Проверьте статус сервера базы данных. Убедитесь, что сервер запущен и функционирует.

1. **База данных не существует:**

Причина: Подключение пытается использовать базу данных, которая не существует.

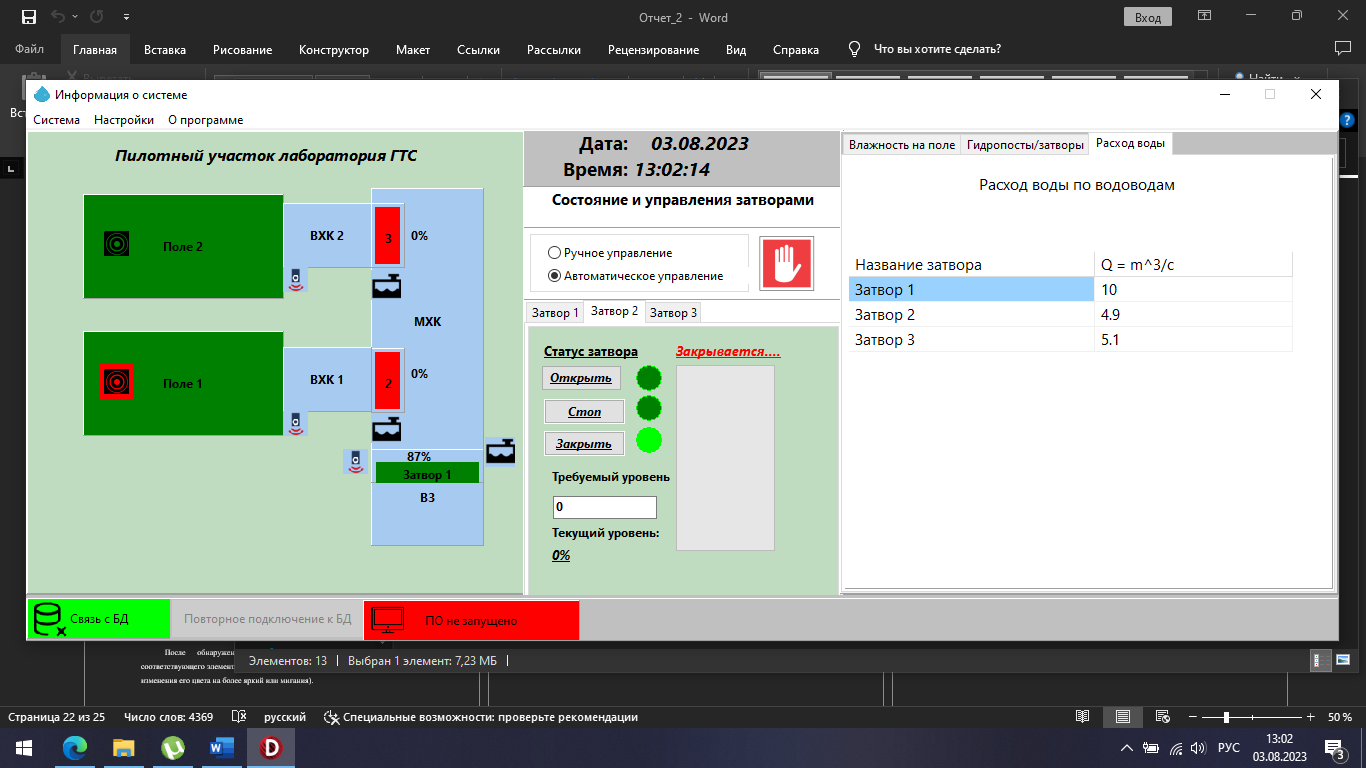
Путь решения: Убедитесь, что указанная база данных существует.

***4.2.2 Зависание программы при обращении к записям базы данных***

В редких случаях происходит зависание самой программы при обращении к записям в базе данных. Причиной такого рода сбоя может быть медленное или нестабильное сетевое соединение между приложением и базой данных что может вызвать долгое ожидание ответа и зависание. В таком случае проверьте состояние сети и доступности сервера с данными. Если хост базы данных находится на удаленном компьютере, проверить доступность сетевого подключения.

***4.2.3 Неправильная калибровка и настройка датчиков влажности***

Неправильная калибровка и настройка датчиков влажности может привести к некорректным данным, что, в свою очередь, может повлиять на точность управления затворами и принятие решений на основе показаний датчиков. При некоторых ошибках датчика система оповестит оператора следующим значком, при наведении на который будет подписана ошибка какого датчика произошла.



Вот некоторые возможные причины неправильной калибровки и настройки датчиков влажности, а также пути их решения:

1. **Неправильная установка:**

Причина: Датчики влажности могли быть неправильно установлены на поле или в окружающей среде.

Путь решения: Перепроверьте правильность установки датчиков. Убедитесь, что они находятся в подходящем месте и не подвержены воздействию факторов, которые могут искажать показания (например, прямое воздействие солнечного света или влаги).

1. **Некорректная калибровка:**

Причина: Датчики влажности могут быть некорректно откалиброваны в процессе настройки.

Путь решения: Потребуется повторно калибровать датчики, тем самым указать минимальные и максимальные показания.

1. **Износ или повреждение датчиков:**

Причина: Датчики влажности могли повредиться или износиться со временем, что может снизить их точность.

Путь решения: Периодически проверяйте состояние датчиков и заменяйте их при необходимости.

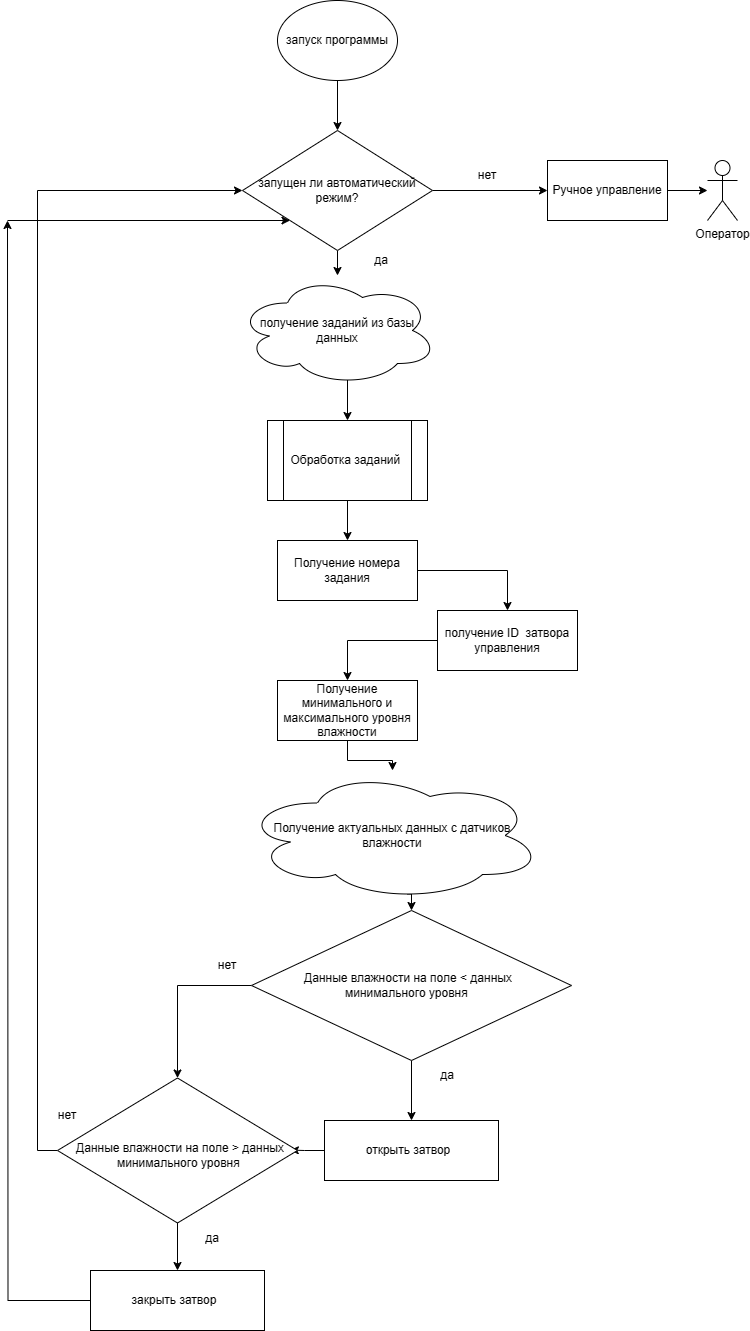
1. **Проблемы с кабелями и подключением:**

Причина: Неправильное подключение датчиков или повреждение кабелей может привести к некорректным показаниям.

Путь решения: Перепроверьте подключение датчиков и состояние кабелей.

Важно тщательно проверить и настроить датчики влажности перед их использованием, а также регулярно следить за их состоянием и точностью. Это поможет обеспечить корректные данные для принятия решений и управления затворами на основе влажности почвы.

# **Графическое представление взаимодействия датчиков влажности с программой**



Ниже представлен словесный вид алгоритма:

1. Начало программы.
2. Запуск программы.
3. Проверка: запущен ли автоматический режим?
4. Если автоматический режим запущен, перейти к шагу 26.
5. Если автоматический режим не запущен:

5.1. Проверка: получены ли задания из базы данных?

5.2. Если задания получены, перейти к шагу 8.

5.3. Если задания не получены, перейти к шагу 9.

1. Ручное управление.
2. Конец программы.
3. Обработка заданий:

8.1. Получение номера задания.

8.2. Получение ID затвора управления.

8.3. Получение минимального и максимального уровня влажности.

8.4. Получение актуальных данных с датчиков влажности.

8.5. Проверка данных влажности на поле:

8.5.1. Если данные больше или равны минимальному уровню, перейти к шагу 11.

8.5.2. Если данные меньше минимального уровня, перейти к шагу 10.

1. Конец программы.
2. Открыть затвор.
3. Проверка данных влажности на поле:

11.1. Если данные меньше или равны максимальному уровню, перейти к шагу 15.

11.2. Если данные больше максимального уровня, перейти к шагу 12.

1. Закрыть затвор.
2. Конец программы.
3. Данные влажности на поле > данных минимального уровня:

14.1. Открыть затвор.

14.2. Проверка данных влажности на поле:

14.2.1. Если данные меньше или равны максимальному уровню, перейти к шагу 18.

14.2.2. Если данные больше максимального уровня, перейти к шагу 15.

1. Закрыть затвор.
2. Конец программы.
3. Данные влажности на поле > данных максимального уровня:

17.1. Закрыть затвор.

17.2. Проверка данных влажности на поле:

17.2.1. Если данные больше максимального уровня, перейти к шагу 20.

17.2.2. Если данные меньше максимального уровня, перейти к шагу 18.

1. Открыть затвор.
2. Конец программы.
3. Данные влажности на поле > данных минимального уровня:

20.1. Открыть затвор.

20.2. Проверка данных влажности на поле:

20.2.1. Если данные меньше или равны максимальному уровню, перейти к шагу 24.

20.2.2. Если данные больше максимального уровня, перейти к шагу 21.

1. Закрыть затвор.
2. Конец программы.
3. Данные влажности на поле > данных максимального уровня:

23.1. Закрыть затвор.

23.2. Проверка данных влажности на поле:

23.2.1. Если данные больше максимального уровня, перейти к шагу 26.

23.2.2. Если данные меньше максимального уровня, перейти к шагу 24.

1. Открыть затвор.
2. Конец программы.
3. Конец программы.

В заключении данной главы описанного алгоритма, можно подытожить следующее:

В данной главе был представлен алгоритм управления системой, который основывается на предварительных настройках и проверках, а также на автоматическом и ручном режимах работы. Алгоритм разрабатывался для обеспечения оптимального контроля за уровнем влажности в заданной области или поле.

Алгоритм начинается с запуска программы и проверки наличия автоматического режима работы. Если автоматический режим активен, то программа переходит к выполнению автоматических заданий. В противном случае, она переходит к ручному управлению.

Важной частью алгоритма является получение заданий из базы данных и обработка этих данных. Для этого система получает номер задания, идентификатор затвора управления, а также минимальный и максимальный уровни влажности. Далее система получает актуальные данные с датчиков влажности.

На основе собранных данных система принимает решение о том, необходимо ли открыть или закрыть затвор для регулирования уровня влажности. Если данные влажности на поле находятся в заданных границах, затвор остается в своем текущем состоянии. В противном случае, система открывает или закрывает затвор в соответствии с заданными условиями.

Таким образом, алгоритм обеспечивает автоматическое управление системой влажности и возможность ручного вмешательства при необходимости. Этот алгоритм может быть применен в различных сферах, где контроль за уровнем влажности играет важную роль в обеспечении оптимальных условий для растений, животных или производственных процессов.

# **ЗАКЛЮЧЕНИЕ**

В заключении можно отметить, что разработанная программа для управления электро-затворами и мониторинга состояния системы на пилотном участке представляет собой эффективное решение для автоматизации процесса орошения. Она позволяет оператору системы управлять затворами и мониторить их состояние, что помогает сократить время реакции на возможные проблемы и повысить эффективность использования водных ресурсов.

Кроме того, использование графической схемы расположения каналов значительно облегчает работу операторов системы, упрощает процесс поиска и локализации проблем в системе, позволяет быстро принимать решения и реагировать на изменения в процессе управления затворами и расходом воды. [1]

Важным преимуществом программы является ее простота использования и гибкость настройки, что позволяет адаптировать ее к конкретным потребностям и требованиям пользователей.

Также следует отметить, что разработанная программа может быть использована в различных условиях и на различных объектах, где применяется система орошения. Она может быть легко адаптирована к конкретным потребностям и требованиям заказчика.

Кроме того, в процессе разработки программы были использованы современные технологии и инструменты, что обеспечивает ее высокую надежность, стабильность и производительность. Были использованы компоненты Delphi, которые обеспечивают широкие возможности для создания графического интерфейса и работы с базами данных.

Таким образом, программа для управления электро-затворами на пилотном участке является эффективным решением для автоматизации системы орошения. Она обеспечивает оператору системы быстрый доступ к информации о состоянии затворов и мониторингу процесса орошения, что повышает эффективность использования водных ресурсов и снижает вероятность возникновения проблем в процессе эксплуатации системы.

В целом, разработка данной программы для управления электро-затворами на пилотном участке является успешным шагом в автоматизации процесса управления затворами и обработки данных в системе орошения. Она предоставляет пользователям удобный и эффективный инструмент для управления затворами и сбора данных, что позволяет повысить эффективность работы системы и оптимизировать использование водных ресурсов.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Д.С. Башкиров, А.А. Воропай, А.Ю. Котов, Н.В. Нестеров. Программно-аппаратный комплекс для автоматизированного управления водными ресурсами оросительных систем // Вестник Самарского государственного технического университета. - 2015. - Том 21, № 2.
2. Жуков Д.В., Жукова Е.А. Информационные технологии в гидрологии: Учебное пособие. - М.: Изд-во МГУ, 2017. - 256 с.
3. Михайлов А.Н., Смирнова Л.А. Инженерия систем управления: Учебное пособие - М.: Изд-во МГУ, 2017. - 256 с.