



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н.П. ОГАРЁВА»  
(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

**ОТЧЕТ**

*о результатах проверки бакалаврской работы обучающегося  
на наличие заимствований*

Автор работы: Сазончик Николай Дмитриевич

Тема работы: Разработка программной части устройства для  
исследования параметров и характеристик цепи управления силовых тиристоров.

Руководитель: Ильин Михаил Владимирович

Представленная работа прошла проверку на наличие заимствований в  
системе «Антиплагиат. ВУЗ»

Результаты автоматической проверки: оригинальность 91,85 %  
цитирования 0,89 %  
заимствования 7,26 %

Результаты анализа полного отчета на наличие заимствований:

правомерные заимствование: да  
корректные цитирования: да  
неправомерные заимствования: нет  
признаки обхода системы: нет

Общее заключение об итоговой оригинальности работы и возможности  
ее допуска к защите: степень оригинальности текста достаточная, работа  
допускается к защите.

Руководитель  
доцент кафедры  
электроники и наноэлектроники



20.06.18 Ильин М.В.

Заявление о самостоятельном характере выполнения  
выпускной квалификационной работы

Я, Сазончик Николай Дмитриевич, студент 4 курса, направления подготовки 12.03.01 «Приборостроение» заявляю, что в моей выпускной квалификационной работе на тему «Разработка программной части устройства для исследования параметров и характеристик цепи управления силовых тиристоров», представленной в Государственную экзаменационную комиссию для публичной защиты, не содержится элементов неправомерных заимствований.

Все прямые заимствования из печатных и электронных источников, а также ранее защищённых письменных работ, кандидатских и докторских диссертаций имеют соответствующие ссылки.

Я ознакомлен с действующим в Университете Положением о проверке выпускных квалификационных работ студентов ФГБОУ ВПО «МГУ им. Н. П. Огарёва» на наличие заимствований, в соответствии с которым обнаружение неправомерных заимствований является основанием для неудовлетворительной оценки выпускной квалификационной работы.

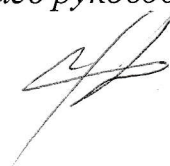
Подпись студента 

Дата 20.06.18

*Работа представлена для проверки в Системе*

*Дата представления ВКР*

*подпись руководителя ВКР*



**ОТЗЫВ**  
**на бакалаврскую работу студента 4-го курса**  
**института электроники и светотехники**  
**направления «Приборостроение» Сазончика Н. Д.,**  
**выполненную на тему:**

**«Разработка программной части устройства для исследования параметров и характеристик цепи управления силовых тиристоров»**


Надёжность преобразователей на основе полупроводниковых приборов во многом зависит от качества изготовления полупроводниковых приборов и подборах их по параметрам в процессе сборки преобразователя. Из-за технологических отклонений наблюдается вариация параметров силовых полупроводниковых приборов. Таким образом, для определения параметров конкретного прибора необходимо иметь соответствующее оборудование. В свою очередь современные компьютерные технологии позволяют использовать программные инструменты для накопления и анализа результатов испытаний. Преимущество данного подхода заключается в том, что вся информация хранится в электронном виде в одном месте и имеется возможность быстро её обработать, что в свою очередь ускоряет процесс подбора полупроводниковых приборов.

В данной работе перед студентом была поставлена задача разработки программного обеспечения испытательного устройства с применением технологий хранения информации на основе базы данных. С данной задачей Н. Д. Сазончик успешно справился. В результате был разработан программный код в среде программирования LabVIEW, который управляет аппаратной частью испытательного устройства, а так же позволяет накапливать и анализировать измерительную информацию в базе данных MySQL. Автором работы были решены задачи настройки обмена данных между аппаратной частью устройства и компьютером, разработана и развернута база данных и созданы инструменты для работы с ней.

Проведенная работа свидетельствует о хорошем уровне знаний Сазончика Н. Д. в области разработки программных средств информационно измерительных систем.

В целом, считаю, что работа Сазончика Николая Дмитриевича заслуживает оценки «отлично», а выпускник Сазончик Н. Д. – присвоения квалификации бакалавра по направлению «Приборостроение».

Руководитель бакалаврской работы  
доцент кафедры  
электроники и нанoeлектроники, к. т. н.


 20.06.18 М. В. Ильин

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н. П. ОГАРЁВА»**


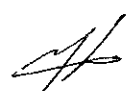

ИНСТИТУТ ЭЛЕКТРОНИКИ И СВЕТОТЕХНИКИ  
КАФЕДРА ЭЛЕКТРОНИКИ И НАНОЭЛЕКТРОНИКИ

УТВЕРЖДАЮ

Заведующий кафедрой  
к.т.н.

 Н. Н. Беспалов  
«22» 06 2018 г.

**БАКАЛАВРСКАЯ РАБОТА  
РАЗРАБОТКА ПРОГРАММНОЙ ЧАСТИ УСТРОЙСТВА ДЛЯ  
ИССЛЕДОВАНИЯ ПАРАМЕТРОВ И ХАРАКТЕРИСТИК ЦЕПИ  
УПРАВЛЕНИЯ СИЛОВЫХ ТИРИСТОРОВ.**

Автор бакалаврской работы	 20.06.2018	Н. Д. Сазончик
Обозначение бакалаврской работы	БР – 02069964 – 12.03.01 – 10 – 18	
Направление подготовки	12.03.01 «Приборостроение»	
Руководитель работы	 20.06.2018	М. В. Ильин
к.т.н., доцент	 20.06.2018	А. А. Шестёркина
Нормоконтролер		

Саранск

2018

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИСЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н. П. ОГАРЁВА»

ИНСТИТУТ ЭЛЕКТРОНИКИ И СВЕТОТЕХНИКИ  
КАФЕДРА ЭЛЕКТРОНИКИ И НАНОЭЛЕКТРОНИКИ

УТВЕРЖДАЮ

Заведующий кафедрой, к.т.н.

 Н. Н. Беспалов

«12» 10 2017 г.

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Студент Сазончик Николай Дмитриевич

1 Тема: Разработка программной части устройства для исследования параметров и характеристик цепи управления силовых тиристоров.

Утверждена приказом по МордГУ № 8292-с от 12.10.2017 г.

2 Срок представления работы к защите 22 июня 2018 г.

3 Исходные данные для выпускной квалификационной работы:

– основное назначение приложения: управление аппаратной частью испытательного устройства; загрузка измерительной информации из аппаратной части; отображение измерительной информации на лицевой панели приложения; сохранение результатов измерения в базе данных; просмотр и поиск результатов измерения в базе данных;

– в приложении должна быть реализована политика безопасности в виде учётных записей операторов;

– база данных должна содержать следующую информацию: учётные записи операторов; библиотеку испытываемых приборов, содержащую паспортные данные; измерительные данные, содержащие оператора, проводившего измерения, дату и время, тип испытываемого прибора, измерительную информацию, комментарий;

– в приложении должны быть предусмотрена возможность формирования отчёта по результатам испытания, просматриваемого в браузере;

– среда разработки приложений – LabVIEW 2012;

– тип базы данных – MySQL;

– программная часть устройства должна быть реализована в виде приложения совместимого с операционной системой Windows 7.

#### 4 Содержание выпускной квалификационной работы.

4.1 Введение.

4.2 Анализ методов и аппаратуры для определения электрических параметров цепи управления силового тиристора.

4.3 Разработка алгоритма верхнего уровня и структуры программного обеспечения.

4.4 Разработка базы данных.

4.5 Разработка инструментов для работы с базой данных.

4.6 Разработка инструментов для формирования отчёта.

4.7 Разработка программного кода для управления аппаратной частью испытательного устройства.

4.8 Заключение.

4.9 Список использованных источников.

4.10 Приложение. Программный код.

Руководитель работы



М. В. Ильин

Задание принял к исполнению

Н. Д. Сазончик

## РЕФЕРАТ

Бакалаврская работа содержит 96 страниц, 64 рисунка, 2 таблицы, 8 источников информации, 1 приложения.

**МИКРОКОНТРОЛЛЕР, БАЗА ДАННЫХ, ФУНКЦИЯ, ИНСТРУМЕНТ, ИНФОРМАЦИЯ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.**

Цель бакалаврской работы – Разработка программной части устройства для исследования параметров и характеристик цепи управления силовых тиристоров.

В процессе работы были рассмотрены различные варианты создания баз данных и программной части для устройства, был выбран наиболее оптимальный вариант для дальнейшей разработки.

В результате бакалаврской работы было разработано программное обеспечение устройства для исследования параметров и характеристик цепи управления силовых тиристоров.

<b>БР - 02069964 - 12.03.01 - 10 - 18</b>								
<i>Лит</i>	<i>Изм.</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>	Разработка программной части устройства для исследования параметров и характеристик цепи управления силовых тиристоров	<i>Лит</i>	<i>Лист</i>	<i>Листов</i>
Разраб.		Н. Д. Сазончик	<i>[Подпись]</i>	20.06.18			4	96
Пров.		М. В. Ильин	<i>[Подпись]</i>	20.06.18				
Т. контр.								
Н. контр.		Шестеркина	<i>[Подпись]</i>	20.06.18				
Утв.		Н.Н. Беспалов	<i>[Подпись]</i>	22.06.18				
<b>МГУ ИЭС ЭНЭ 451</b>								



## Сокращение слов

Таблица 1 — Сокращения часто употребляемых слов

Слово (Словосочетание)	Сокращение
Интегральная микросхема	ИС
Персональный компьютер	ПК
База данных	БД

## СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ.	7
1 Анализ методов и аппаратуры для определения электрических параметров цепи управления силового тиристора.	8
1.1 Комплекс «АДИП».	8
1.2 Комплекс «ИПЭУ-2».	9
2 Разработка алгоритма верхнего уровня и структуры программного обеспечения.	12
3 Разработка базы данных.	15
4 Разработка инструментов для работы с базой данных.	18
4.1 Функция для подключения к базе данных.	19
4.2 Разработка инструментов для записи.	21
4.2.1 Запись пользователя в базу данных.	21
4.2.2 Запись тиристора в базу данных.	24
4.2.3 Запись измерения в базу данных.	28
4.3 Разработка инструментов для чтения.	31
4.3.1 Чтение записей по пользователям из базы данных.	31
4.3.2 Чтение записей по тиристорам из базы данных.	34
4.3.3 Чтение записей по измерениям из базы данных.	37
4.4 Разработка инструмента для формирования запроса в базу.	40
5 Разработка инструментов для формирования отчёта.	45
6 Разработка программного кода для управления аппаратной частью испытательного устройства.	48
6.1 Инструмент для снятия измерений.	48
6.2 Программирование контроллера.	50
ЗАКЛЮЧЕНИЕ.	57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.	58
ПРИЛОЖЕНИЕ А (обязательно) программный код.	59

## ВВЕДЕНИЕ

На данный момент идет активное развитие измерительных приборов. Однако, вывод информации и запись измерений проводится довольно неудобным способом.

В последнее время все больше измерительных приборов связывают с ПК для удобства работы с ним. Благодаря огромному развитию программного обеспечения появилось много программ которые способны работать с измерительными приборами и обрабатывать информацию, полученную с них. Преимуществами работы с ПК является довольно простая и понятная визуализация результатов измерения, возможность проводить какие-либо вычисления с измерениями, сохранять измерения с измерительных приборов. И для сохранения информации используют базу данных

База данных – представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины. Программы для создания баз данных намного упростили обработку каких-либо видов информации.

Совокупность, программ для работы с измерительными устройствами и баз данных позволяет создать, довольно простое в обращении и с большими возможностями по обработке информации, программное обеспечение для прибора.

									Лист
									7
Изм.	Лист	№ докум.	Подпись	Дата	БР-02069964-12.03.01-10-18				

# 1 Анализ методов и аппаратуры для определения электрических параметров цепи управления силового тиристора

Рассмотрим способ вывода информации. Ниже будут приведены примеры устройств для определения параметров тиристоров

## 1.1 Комплекс «АДИП»

Научно-производственное предприятие «Электронная техника – МГУ», организованное при Мордовском государственном университете им. Н. П. Огарёва, производит испытательное оборудование серии «АДИП» (рисунок 1).

Оборудование серии «АДИП» позволяют измерять основные параметры и характеристики силовых полупроводниковых приборов.



Рисунок 1 — Внешний вид комплекса «АДИП»

В комплекс «АДИП» входит четыре установки. Все установки соответствуют требованиям ГОСТ 24461 – 80. Результаты измерения выводятся на цифровую индикацию. Масса не превышает 35кг, а габариты 750х600х600 мм.

«АДИП-1» предназначен для измерения значений импульсных токов в закрытом (обратном) состояниях  $I_{DRM}$  ( $I_{RRM}$ ) силовых полупроводниковых приборов в пределах от 10 мкА до 200 мА при заданном классе по напряжению в пределах от 300 В до 9000 В.

«АДИП-2» применяется для измерения импульсных напряжений в открытом (прямом) состоянии  $U_{TM(FM)}$  всех современных типов силовых полупроводниковых приборов. Устройство позволяет формировать через силовые полупроводниковые приборы испытательные однократные импульсы тока с амплитудой  $I_{TM(FM)}$  в пределах от 10 А до 15 кА и измерять  $U_{TM(FM)}$  в пределах от 0,5 В до 10 В.

«АДИП-3» служит для измерения времени задержки  $t_{gd}$  и времени включения  $t_{gt}$  всех известных типов силовых тиристоров и силовых симисторов в пределах от 0,5 мкс до 30 мкс.

«АДИП-4» предназначен для измерения параметров цепи управления силовых тиристоров и силовых симисторов отпирающего тока  $I_{GT}$  в пределах от 10 мА до 1 А и отпирающего напряжения  $U_{GT}$  в пределах от 0,5 В до 10 В.

В установках используется защита испытуемого СПП от разрушения во время испытаний.

Основным недостатком оборудования серии «АДИП» является отсутствие возможности взаимодействия с ПК, что снижает функциональность и универсальность.

## **1.2 Измеритель параметров управляющих электродов силовых тиристоров «ИПЭУ-2»**

Измеритель ИПЭУ-2 представляет собой переносное устройство с сетевым кабелем и комплектом проводов для подключения к проверяемому силовому тиристор, управляющей ПЭВМ (при работе в составе автоматизированного комплекса) (рисунок 2).

					<b>БР-02069964-12.03.01-10-18</b>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9



Рисунок 2 — Внешний вид «ИПЭУ-2»

Измеритель предназначен для определения величин отпирающего постоянного тока управления ( $I_{GT}$ ) и отпирающего постоянного напряжения управления ( $U_{GT}$ ) силовых тиристоров (СТ).

Измеритель позволяет определять параметры цепи управления тиристорами отпирающего тока в пределах 10 – 500 мА и отпирающего напряжения в пределах 0,5 – 8 В.

Проверяемые параметры:

- отпирающий постоянный ток ( $I_{GT}$ ) силовых тиристоров;
- отпирающее постоянное напряжение ( $U_{GT}$ ) силовых тиристоров

Технические характеристики:

Таблица 2 – параметры «ИПЭУ-2».

Наименование параметра	Значение
Напряжение питания, В	220
Потребляемая мощность в режиме измерения, Вт	не более 150
Диапазон измерения отпирающего тока $I_{GT}$ управления силовыми тиристорами мА	от 10 до 500

Продолжения таблицы 2.

Диапазон измерения отпирающего постоянного напряжения $U_{GT}$ управления В;	от 0 до 8,0
Индикация тока $I_{GT}$ силового тиристора	цифровая с количеством разрядов – 4
Индикация напряжения $U_{GT}$ силового тиристора	цифровая с количеством разрядов 4
Погрешность измерений тока $I_{GT}$ и напряжения $U_{GT}$ , %	не более $\pm 10$
Индикация о неисправном состоянии силового тиристора или цепи управления силовым тиристором	светодиодная
Автоматическое управление измерителем осуществляется по стандарту	EIA RS-485

Основной особенностью электрической части измерителя является возможность определения отпирающего тока и отпирающего напряжения силовых тиристоров, в ручном и в автоматизированном режиме (в составе автоматизированного комплекса для определения параметров силовых полупроводниковых приборов КАИСПП).

## 2 Разработка алгоритма верхнего уровня программного обеспечения

Была разработана функциональная схема для наглядного представления, всех процессов, которые проходят в приборе (рисунок 3).

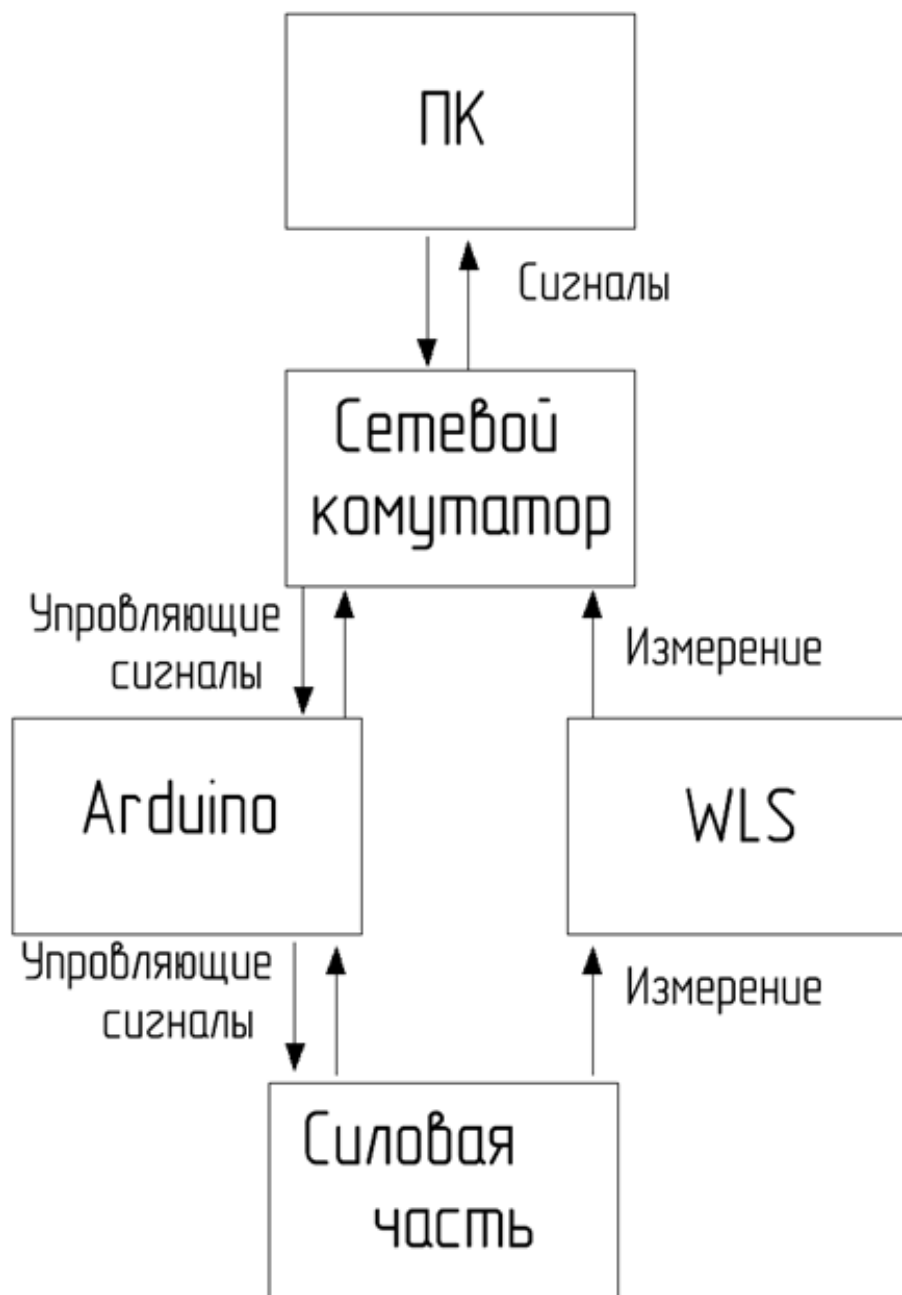


Рисунок 3 – Функциональная схема в прибора

Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

12



С силовой части прибора будут приходить несколько сигналов. Первый сигнал который будет снимать WLS ,будет отвечать за измерения. Вторым сигналом с Arduino будет управлять режимом силовой части. Он будет устанавливать силовую часть в режим измерения при котором будет возможность снять измерения с тиристора, вторым режимом будет отвечать за нахождение силовой части в режиме ожидания. В режиме ожидания будет возможность безопасно отключить резистор. Вся информация будет проходить через сетевой коммутатор передачи информации и с Arduino и с WLS на ПК.

Составить верхний алгоритм программного обеспечения. Алгоритм будет показывать основные этапы работы программы для того чтобы снять измерения (рисунок 4).

В программе будет возможность работы с информацией сохранённой в базе. Пользователь будет иметь возможность сохранять новые записи по пользователю, тиристорам и измерениям. Так же в программе будет возможность вывода информации с базы и формирование по ней отчета.

					<b>БР-02069964-12.03.01-10-18</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		13

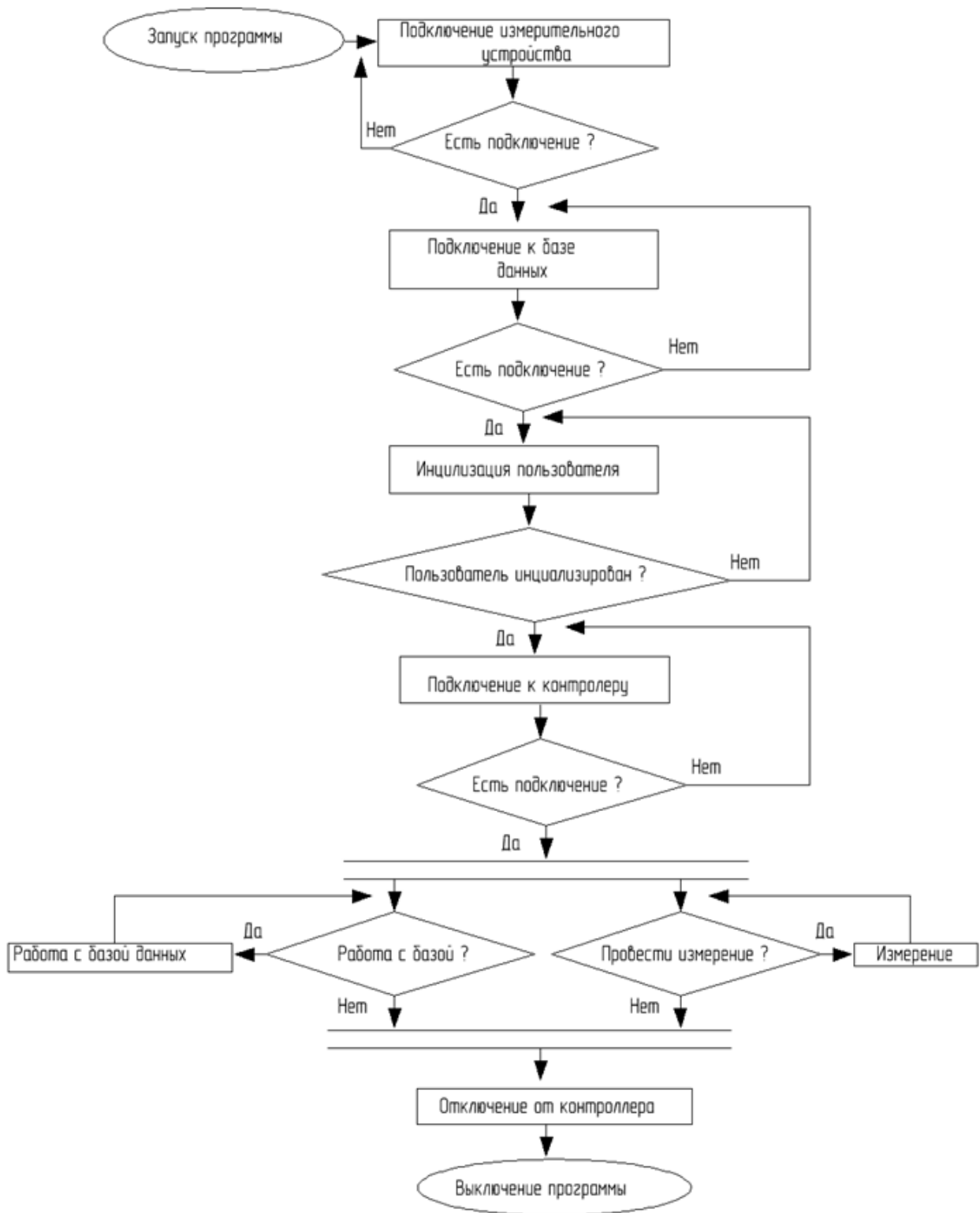


Рисунок 4 – Алгоритм программы

### 3 Разработка базы данных

Для создания базы данных (БД) была выбрана программа MySQL по причинам ее удобства и знакомства с ней. Первым этапом по разработке БД стало, создание модели в программе MySQL (рисунок 5).

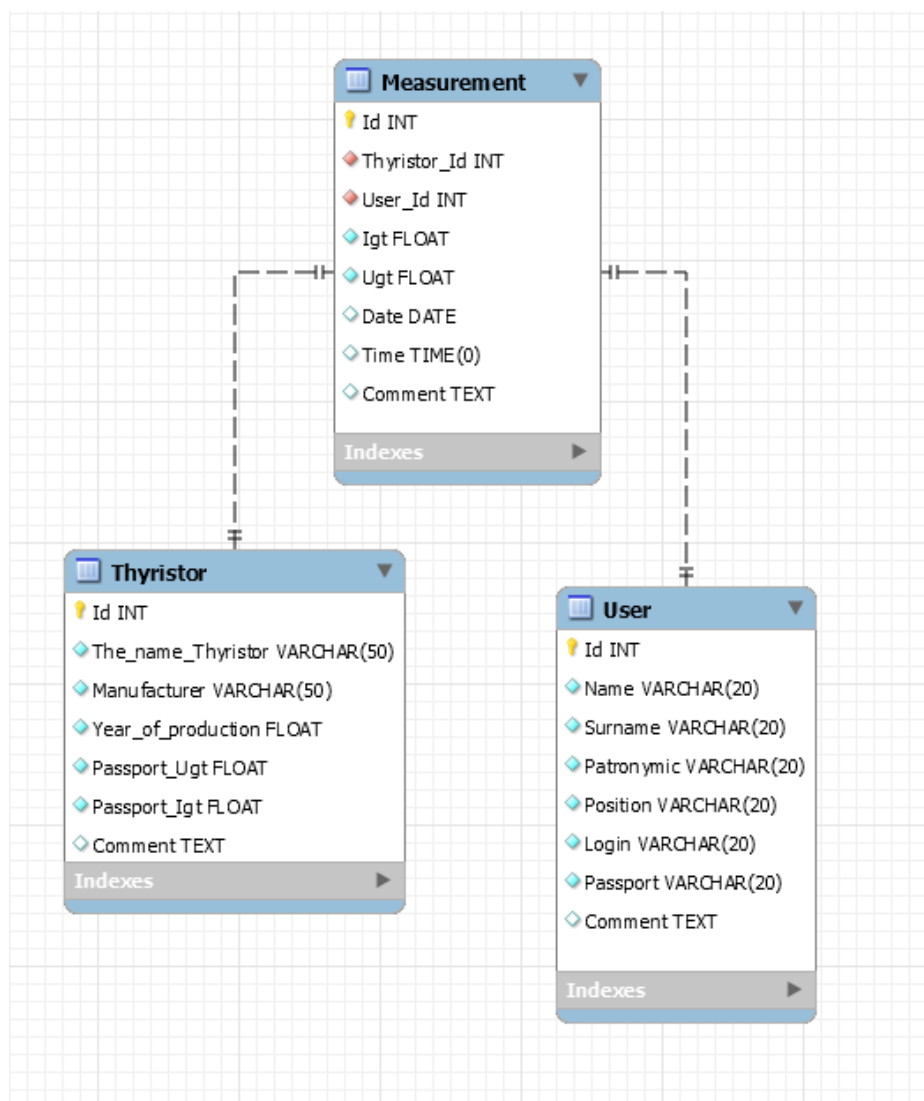


Рисунок 5 – Модель базы данных в программе MySQL

В БД будут входить таблица «Measurement» предназначенная для записей измерительной информации и данных об измерениях, таблица «Thyristor» содержит информацию о тиристоре и его паспортных параметрах, таблица «User» содержит данные учётных записей операторов.

В таблицу «Thyristor» входят следующие поля.

**ID** содержит уникальный идентификационный номер типа тиристора. Тип данных этого поля был выбран INT. Поле заполняется автоматически при создании новой записи.

**The\_name\_Thyristor** содержит название тиристора. Тип данных этого поля был выбран VARCHAR(50).

**Manufacturer** содержит наименование компании производителя тиристора. Тип данных этого поля был выбран VARCHAR(50).

**Year\_of\_production** содержит год производства тиристора. Тип данных был выбран DATE.

**Pasport\_Ugt** содержит паспортное значение напряжения  $U_{gt}$  тиристора. Тип данных был выбран FLOAT.

**Pasport\_Igt** содержит паспортное значение тока  $I_{gt}$  тиристора. Тип данных был выбран FLOAT.

**Comment** содержит дополнительную информацию, необходимую для идентификации тиристора. Тип данных был выбран TEXT. Заполнение данного поля не является обязательным.

В таблицу «User» входят следующие поля.

**ID** содержит уникальный идентификационный номер учётной записи оператора. Тип данных этого поля был выбран INT. Будет заполняться автоматически при создании новой учётной записи.

**Name** содержит имя оператора. Тип данных этого поля был выбран VARCHAR(20).

**Surname** содержит фамилию оператора. Тип данных этого поля был выбран VARCHAR(20).

**Patronymic** содержит отчество оператора. Тип данных был выбран VARCHAR(20).

**Login** содержит логин оператора. Тип данных этого поля был выбран VARCHAR(20).

**Password** содержит пароль оператора. Тип данных этого поля был выбран VARCHAR(20).

**Position** содержит должность оператора на предприятии. Тип данных был выбран VARCHAR(20).

**Comment** содержит какую-либо дополнительную информацию об операторе. Тип данных был выбран TEXT.

В таблицу «Measurement» входят следующие поля.

**ID** содержит уникальный идентификационный номер измерения. Тип данных этого поля был выбран INT. Поле заполняется автоматически при создании новой записи.

**Thyristor\_Id** содержит уникальный идентификационный номер типа испытуемого тиристора. Тип данных этого поля был выбран INT.

**User\_Id** содержит уникальный идентификационный номер оператора, который производил измерение. Тип данных этого поля был выбран INT.

**Igt** содержит значение тока  $I_{gt}$ , полученное в результате измерения. Тип данных был выбран FLOAT.

**Ugt** содержит значение напряжения  $U_{gt}$ , полученное в результате измерения. Тип данных был выбран FLOAT.

**Date** содержит дату выполнения измерения. Тип данных был выбран DATE.

**Time** содержит время выполнения измерения. Тип данных был выбран TIME(0).

**Comment** содержит дополнительную информацию, поясняющую результаты измерений. Тип данных был выбран TEXT. Заполнение поля не обязательное.

После создания модели мы формируем с помощью функции «Forward engineer» базу данных.

## 4 Разработка инструментов для работы с базой данных

Для начала работы с базой данных первым шагом мы должны установить связь с ней. Это достигается с помощью программы `odbcad32`. В поле «Data Source Name» необходимо вбить имя для связи, мы используем «thyristor». Вкладку «Description» мы пропускаем из-за того, что мы не нуждаемся в описании. В вкладку «TCP/IP» мы вбиваем «localhost» из-за того, что база находится на компьютере, где проводится работа программы. В вкладку «User» мы вбиваем довольно легкое название «root» для облегчения работы. В вкладку «Password» мы так же вбиваем вкладку «root» для облегчения работы. В вкладке «Database» выбираем имя созданной базы данных (рисунок 6). Нажимаем кнопку «OK», тем самым создаем связь для нашей созданной базы данных.

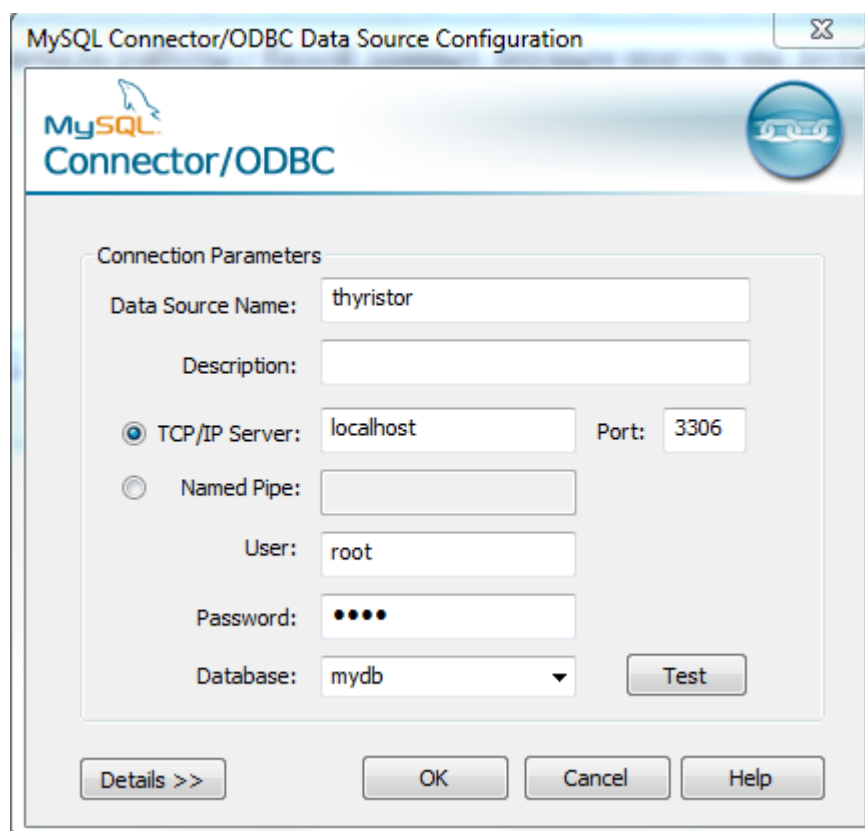


Рисунок 6 – Настройка связи с базой данных

Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

18

## 4.1 Функция для подключения к базе данных

Запись в базу данных заключается в несколько этапах. Первым этапом по созданию функций, работающих с базой данной, служит подключение к ней. Для того что бы подключиться мы используем в программе LabVIEW функцию «DB Tools Open Connec».

На вход «userID» мы должны подать название пользователя связи, в данном случаи «root», на вход «connection information» мы должны подать имя связи с базой данных в данном случаи «thyristor» и последним важным пунктом для подключения к базе данных является подача на вход «password» пароля который мы указали при создание связи «root». Остальные входы не играют важную роль в работе с базой данных. В итоге мы получили такую схему подключения (рисунок 7).

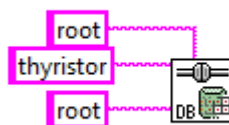


Рисунок 7 – Схема подключения функции DB Tools Open Connec

Подключение будет производиться в каждой функции из-за того, что без этого нельзя будет работать с базой данных.

Из за того что программа делается в классе функция подключения к базе данных будет сделана подклассом, сначала будет сделана функция для загрузки информации для подключения в класс (рисунок 8).

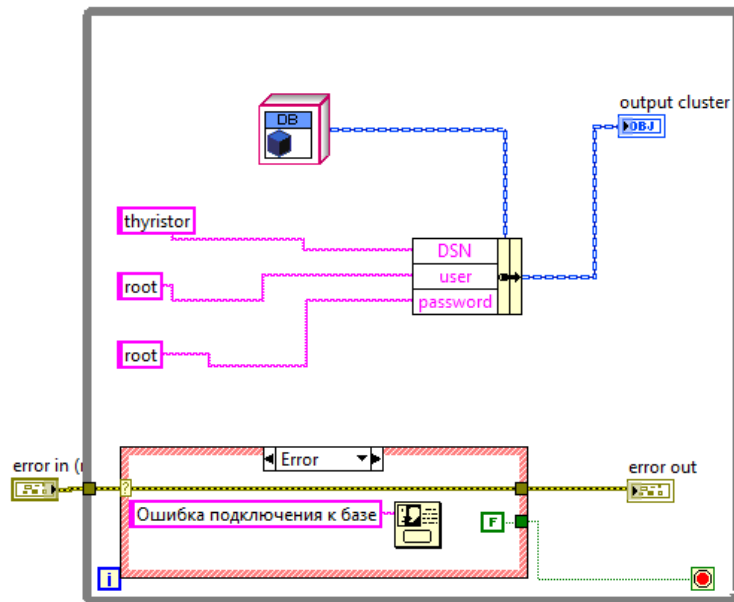


Рисунок 8 – Программа для внесения информации подключения в класс

Для уменьшения массива, программа для внесения информации дальше будет использоваться как функция с иконкой (рисунок 9).



Рисунок 9 – Функция для внесения информации подключения в класс

Так же функция подключения будет переделана в подкласс (рисунок 10)

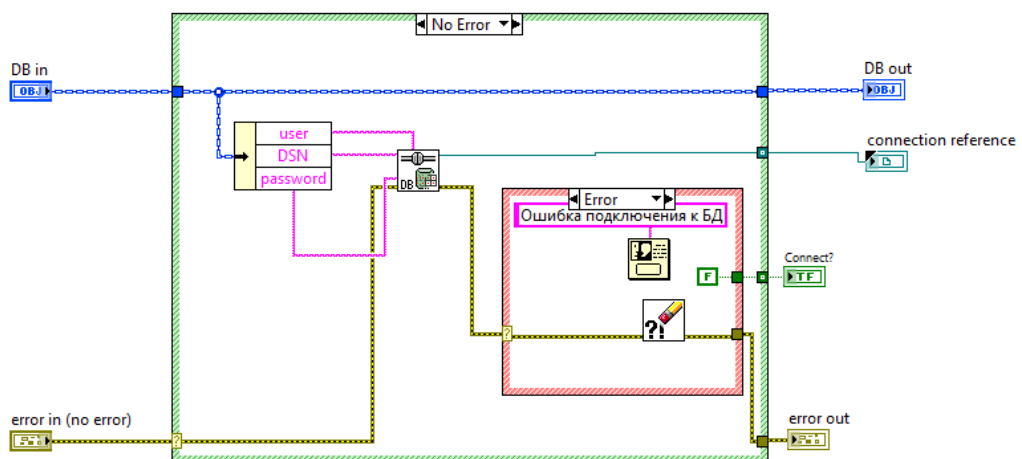


Рисунок 10 – Программа для подключения к базе данных



Для уменьшения массива, программа для подключения к базе дальше будет использоваться как функция с иконкой (рисунок 11).



Рисунок 11 – Иконка функции для подключения

## 4.2 Разработка инструментов для записи

### 4.2.1 Запись пользователя в базу данных

Запись в базу данных заключается в несколько этапах. Первым этапом служит подключение к базе. Дальше для работы с базой данных мы берем функцию для записи информации «DB Tools Insert Data VI».

На вход «data» мы должны подать информацию которую мы хотим записать в базу данных ( контролеры для записей должны стоять именно в таком порядке в котором были заявлены столбцы в таблице базы данных), на вход " connection information" мы должны провести линию связи с функцией «DB Tools Open Connec», на вход «table» мы подаем название таблицы в базе данных с которой мы хотим работать «User», на вход « columns» мы должны подать массив тех столбцов таблицы в которые мы хотим записать информацию и последним обязательным пунктом является проведение линии ошибки на вход «error in». Завершающим пунктом является отключение от базы данных с помощью функции «DB Tools Clous Connection VI». В итоге мы получили полную сформированную схему для записи пользователя в базу данных (рисунок 12).

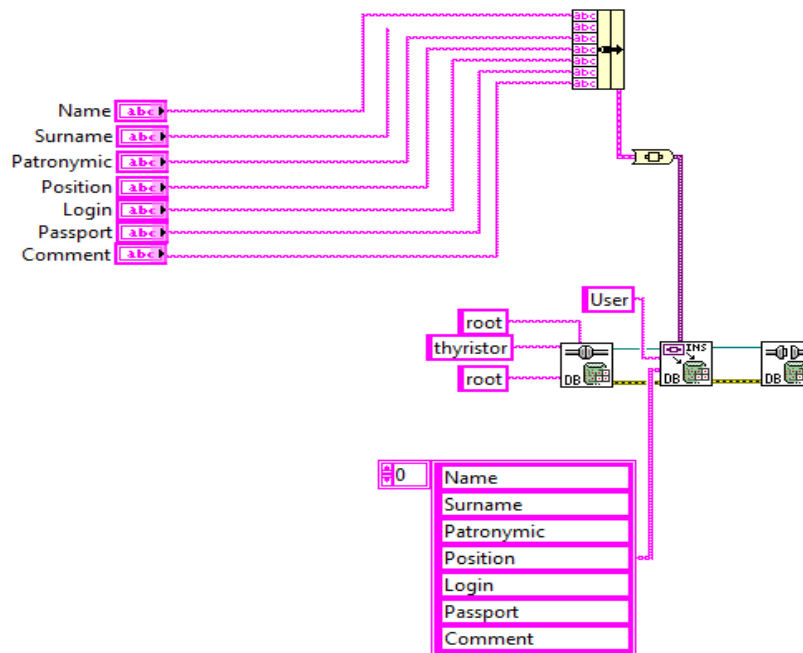


Рисунок 12 – Полная схема по записи пользователя в базу

После окончания схемы по записи пользователя, необходимо проверить её работоспособность. В базе данных было сделано две записи для того, чтобы было наглядно видно как работает схема (рисунок 13).

1 • `SELECT * FROM mydb.user;`

Id	Name	Surname	Patronymic	Position	Login	Passport	Comment
1	Nikolay	Sarov	Nikolayevich	Student	Nikolay	123	comment 1
2	Sergey	Kamarov	Sergeevich	Teacher	Ser	456	comment 2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 13 – Записи в базе данных

Теперь попробуем записать нового пользователя (рисунок 14). Разные языки специально используются для того, чтобы показать, что программа может записывать на разных языках.

Имя

Фамилия

Отчество

Логин

Пароль

Должность

Оставьте свой комментарий

Рисунок 14 – Запись нового пользователя

Теперь снова задаем запрос в базе данных на вывод таблицы "User" (рисунок 15).

```
1 • SELECT * FROM mydb.user;
```

	Id	Name	Surname	Patronymic	Position	Login	Passport	Comment
▶	1	Nikolay	Sarov	Nikolayevich	Student	Nikolay	123	comment 1
	2	Sergey	Kamarov	Sergeevich	Teacher	Ser	456	comment 2
	3	Максим	Лисин	Николаевич	Инженер	Lis	567	coment 3
✖	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 15 – Записи в базе данных

Как мы видим в базе данных пользователя появилась новая запись, а это значит, что схема по записи нового пользователя работает исправно.

Перед использованием в основной программе программа была переделана (рисунок 16).

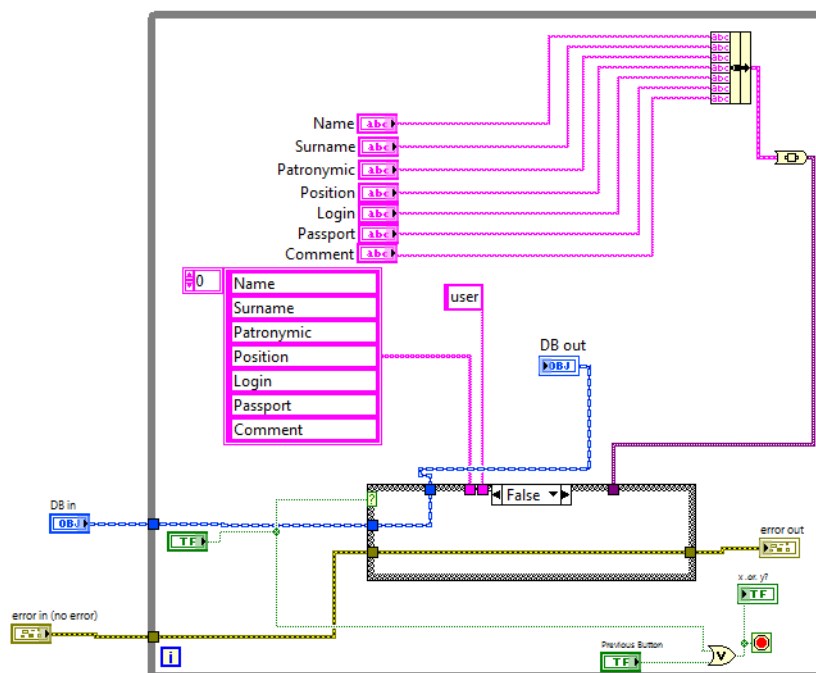


Рисунок 16 – Законченная функция по записи данных по пользователю

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок17).



Рисунок 17 – Иконка функции для записи данных по пользователю

#### 4.2.2 Запись тиристора в базу данных

Запись тиристора в базу данных проводится в несколько этапов. Первым является подключение к базе данных, дальше для записи в базу данных мы берем функцию «DB Tools Insert Data VI».

На вход «data» мы должны подать информацию которую мы хотим записать в базу данных ( контролеры для записей должны стоять именно в таком порядке в котором были заявлены столбцы в таблице базы данных), на

вход " connection information" мы должны провести линию связи с функцией «DB Tools Open Connec», на вход «table» мы подаем название таблицы в базе данных с которой мы хотим провести работу, в этом случае это «Thyristor», на вход « columns» мы должны подать массив тех столбцов таблицы в которые мы хотим записать информацию и последним обязательным пунктом является проведение линии ошибки на вход «error in». Завершающим этапом является отключение от базы с помощью функции «DB Tools Clous Connection VI». В итоге мы получили такую схему для записи тиристора в базу данных (рисунок 18).

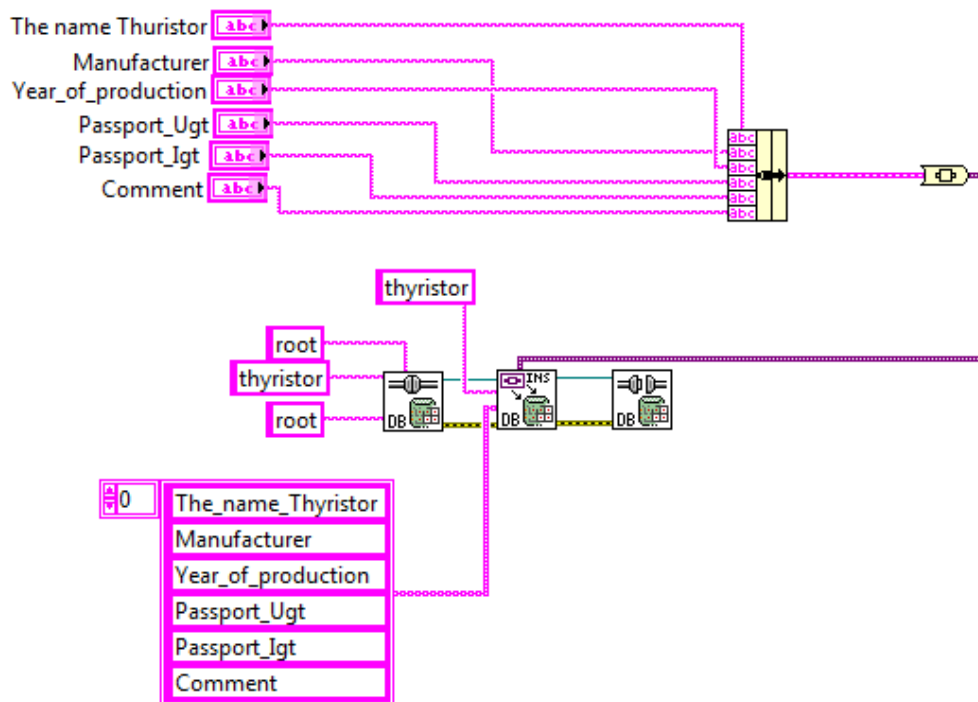


Рисунок 18 – Полная схема для записи тиристора в базу

После завершения схемы для записи тиристора, необходимо проверить её работоспособность. В базе данных было сделано две записи для того, чтобы было наглядно видно как работает схема (рисунок 19).

```
1 • SELECT * FROM mydb.thyristor;
```

	Id	The_name_Thyristor	Manufacturer	Year_of_production	Passport_Ugt	Passport_Igt	Comment
▶	1	T122-25-1	NXP Semiconductors	2015	60	2.5	comment 1
	2	T142-50-13	Unisonic Technologies	2011	120	3	comment 2
*		NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 19 – Записи в базе данных

Теперь попробуем записать новый тиристор (рисунок 20). Разные языки специально используются для того, чтобы показать, что программа может записывать на разных языках.

Название теристора

Производитель

Год производства

Паспортное Ugt

Паспортное Igt

Комментарий

Рисунок 20 – Запись нового тиристора

Теперь производим еще один запрос в базе данных для того, чтобы вывести таблицу «thyristor» (рисунок 21).

```
1 • SELECT * FROM mydb.thyristor;
```

	Id	The_name_Thyristor	Manufacturer	Year_of_production	Passport_Ugt	Passport_Igt	Comment
▶	1	T122-25-1	NXP Semiconductors	2015	60	2.5	comment 1
	2	T142-50-13	Unisonic Technologies	2011	120	3	comment 2
	3	T122-48-4	NXP Semiconductors	2013	70	1.3	comment3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 21 – Записи в базе данных

Перед использованием в основной программе программа была переделана и сделана подклассом (рисунок 22).

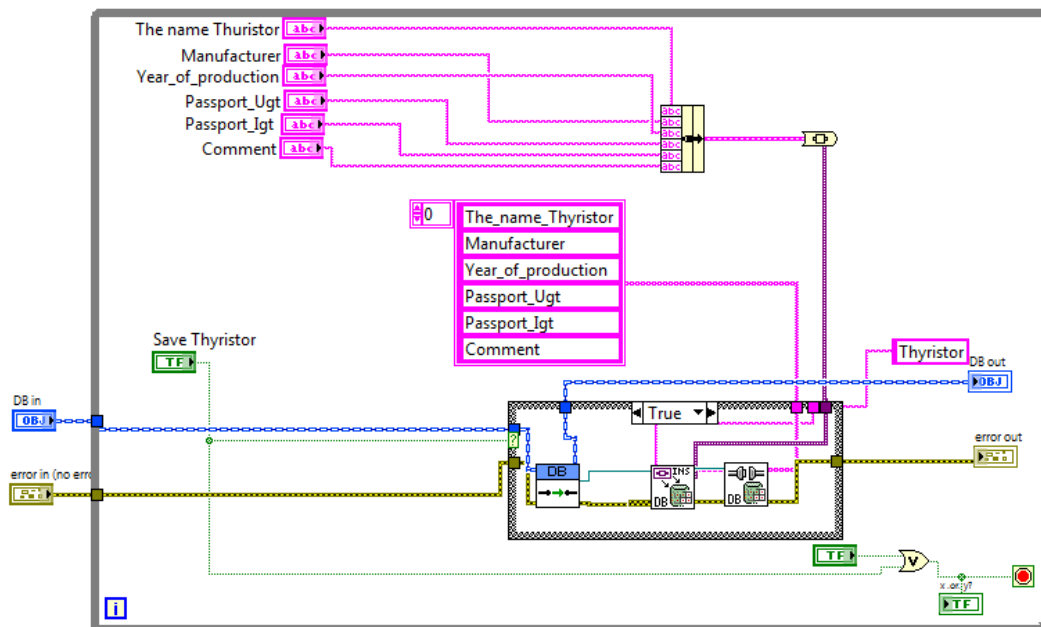


Рисунок 22 – Законченная функция по записи данных по тиристор

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок 23).



Рисунок 23 – Иконка функции для записи данных по тиристор

### 4.2.3 Запись измерения в базу данных

Запись тиристора в базу данных проводится в несколько этапов. Первым является подключение к базе данных, дальше для записи в базу данных мы берем функцию «DB Tools Insert Data VI».

На вход «data» мы должны подать информацию которую мы хотим записать в базу данных ( контролеры для записей должны стоять именно в таком порядке в котором были заявлены столбцы в таблице базы данных), на вход " connection information" мы должны провести линию связи с функцией «DB Tools Open Connec», на вход «table» мы подаем название таблицы в базе данных с которой мы хотим работать в этом случае это «Measurement», на вход «columns» мы должны подать массив тех столбцов таблицы в которые мы хотим записать информацию и последним обязательным пунктом является проведение линии ошибки на вход «error in». Завершающим этапом является отключение от базы с помощью функции «DB Tools Clous Connection VI». В итоге мы получили такую схему для записи тиристора в базу данных (рисунок 24).

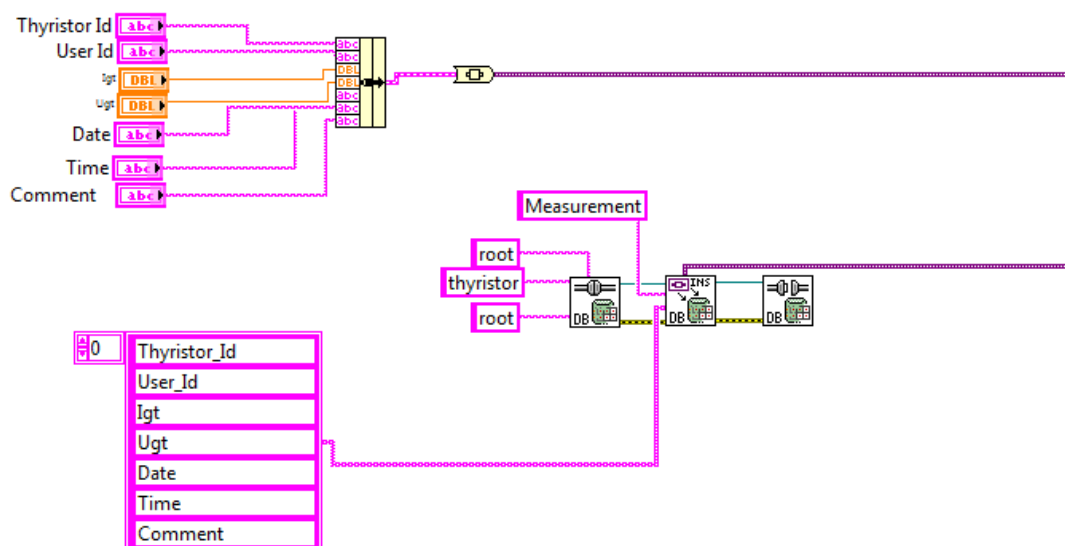


Рисунок 24 – Полная схема для записи измерений



После завершения схемы для записи измерений в базу данных, необходимо проверить её работоспособность. В базе данных было сделано две записи для того, чтобы было наглядно видно, как работает схема (рисунок 25).

The screenshot shows a SQL query window with the command: `SELECT * FROM mydb.Measurement;` Below the query is a 'Result Grid' containing two rows of data. The columns are: Id, Thyristor\_Id, User\_Id, Igt, Ugt, Date, Time, and Comment. The first row has values: 1, 2, 2, 0.9, 20, 2017-05-14, 15:16:17, and NULL. The second row has values: 2, 1, 1, 1.2, 10, 2018-04-15, 16:14:27, and NULL.

Id	Thyristor_Id	User_Id	Igt	Ugt	Date	Time	Comment
1	2	2	0.9	20	2017-05-14	15:16:17	NULL
2	1	1	1.2	10	2018-04-15	16:14:27	NULL

Рисунок 25 – Записи в базе данных

Теперь попробуем записать новое измерение (рисунок 26). Разные языки специально используются для того, чтобы показать, что программа может записывать на разных языках.

The screenshot shows a form with the following fields and values: Thyristor Id: 1, User Id: 2, Igt: 1,3, Ugt: 30, Date: 2018-05-21, Time: 16:18:20, and Comment: coment1.

Рисунок 26 – Запись нового измерения

Теперь производим еще один запрос в базе данных для того, чтобы вывести таблицу «Measurement» (рисунок 27).

```
1 • SELECT * FROM mydb.Measurement;
```

	Id	Thyristor_Id	User_Id	Igt	Ugt	Date	Time	Comment
▶	1	2	2	0.9	20	2017-05-14	15:16:17	NULL
	2	1	1	1.2	10	2018-04-15	16:14:27	NULL
	3	1	2	1.3	30	2018-05-21	16:18:20	coment1

Рисунок 27 – Записи в базе данных

Как мы видим в базе данных измерений появилась новая запись, а это значит, что функция по записи нового тиристора работает исправно.

Перед использованием в основной программе программа была переделана и сделана подклассом (рисунок 28).

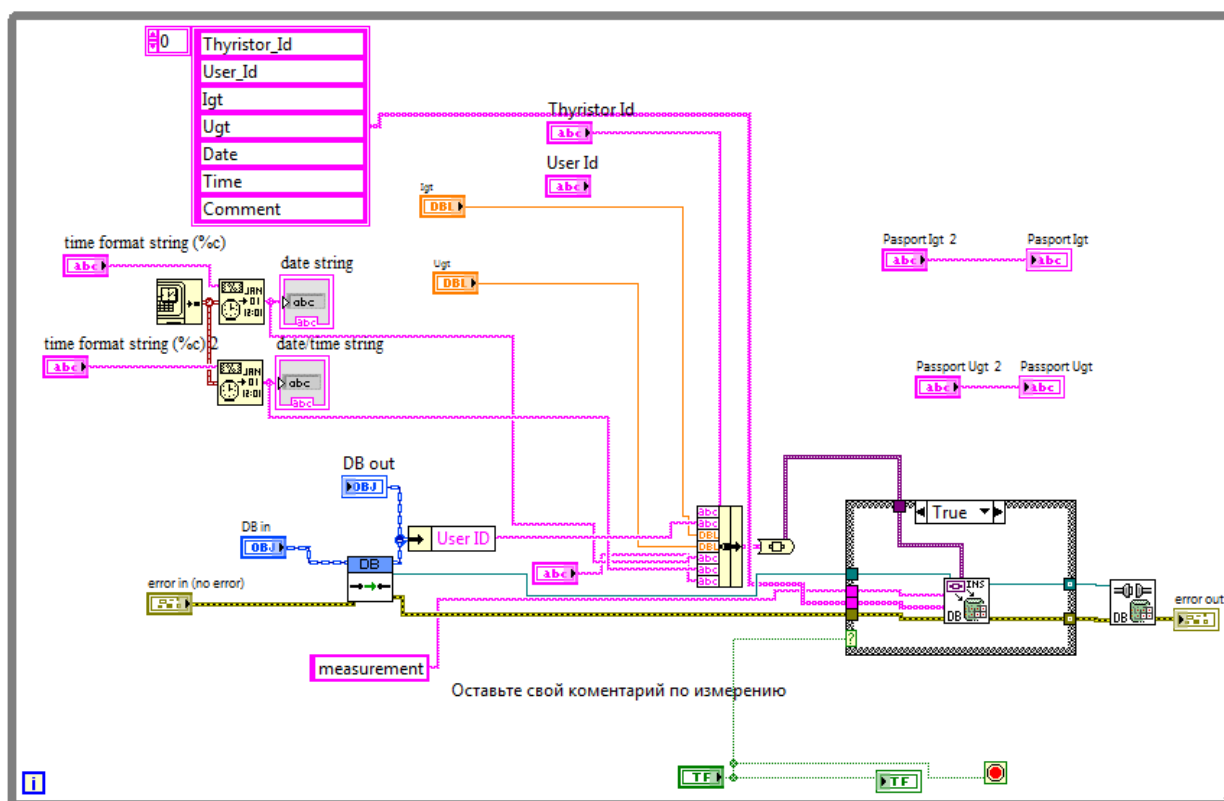


Рисунок 28 – Законченная функция по записи данных по измерению

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок 29).



Рисунок 29 – Иконка функции для записи данных по тиристор

### 4.3 Разработка инструментов для чтения

#### 4.3.1 Чтение записей по пользователям из базы данных

Чтение из базы данных производится в несколько этапов. Первым этапом служит подключение к базе данных. Далее для работы с базой данных мы берем функцию для записи информации «DB Tools Select Data».

На вход «connection information» мы должны провести линию связи с функцией «DB Tools Open Connec», на вход «table» мы подаем название таблицы в базе данных с которой мы хотим работать в этом случае это «User», если мы хотим вывести определенные столбцы таблицы то на вход «columns» мы должны подать в формате массив те столбцы которые мы хотим считать с базы данных и последним обязательным пунктом является проведение линии ошибки на вход «error in». В итоге мы получили такую схему подключения (рисунок 30).

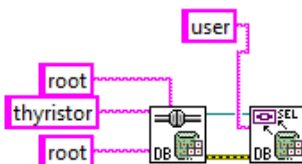


Рисунок 30 – Схема подключения функции DB Tools Select Data

После запроса в базу данных на вывод таблицы «User» нужно ее вывести. Для этого мы используем функцию «Database Variant To Data».

На вход «database variant» мы должны провести линию связи с выхода «date» функции «DB Tools Select Data», на вход «type» мы подаем тип, в котором нужно выводить информацию, в данном случае это массив, на выход «date» мы ставим индикатор. Завершающим этапом является отключение от базы с помощью функции «DB Tools Clous Connection VI». В итоге мы получили такую схему подключения (рисунок 31).

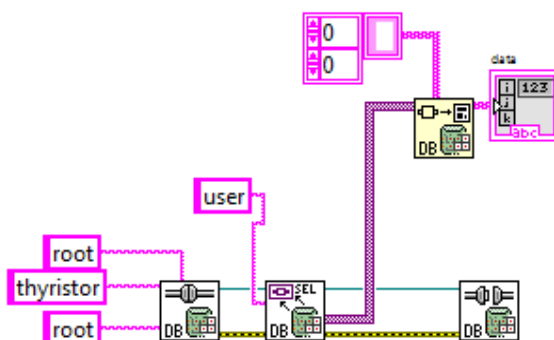


Рисунок 31 – Полная схема для чтения записей по пользователям

После завершения схемы для чтения записей по пользователям, необходимо проверить её работоспособность. В базе данных было сделано две записи для того, чтобы было наглядно видно, как работает схема (рисунок 32).

1 • `SELECT * FROM mydb.user;`

Id	Name	Surname	Patronymic	Position	Login	Passport	Comment
1	Nikolay	Sarov	Nikolayevich	Student	Nikolay	123	comment 1
2	Sergey	Kamarov	Sergeevich	Teacher	Ser	456	comment 2
* NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 32 – Записи в базе данных

После запуска программы на индикатор выводиться информация, идентичная информации, которая у нас записана в базе данных, это значит, что функция по чтению измерений работает исправно (рисунок 33).

data								
0	1	Nikolay	Sarov	Nikolayevich	Student	Nikolay	123	comment 1
0	2	Sergey	Kamarov	Sergeevich	Teacher	Ser	456	comment 2

Рисунок 33 – Индикатор для вывода информации с функции Database Variant To Data

Перед использованием в основной программе программа была переделана и сделана подклассом. Нижний цикл был сделан для формирования отчета (рисунок 34).

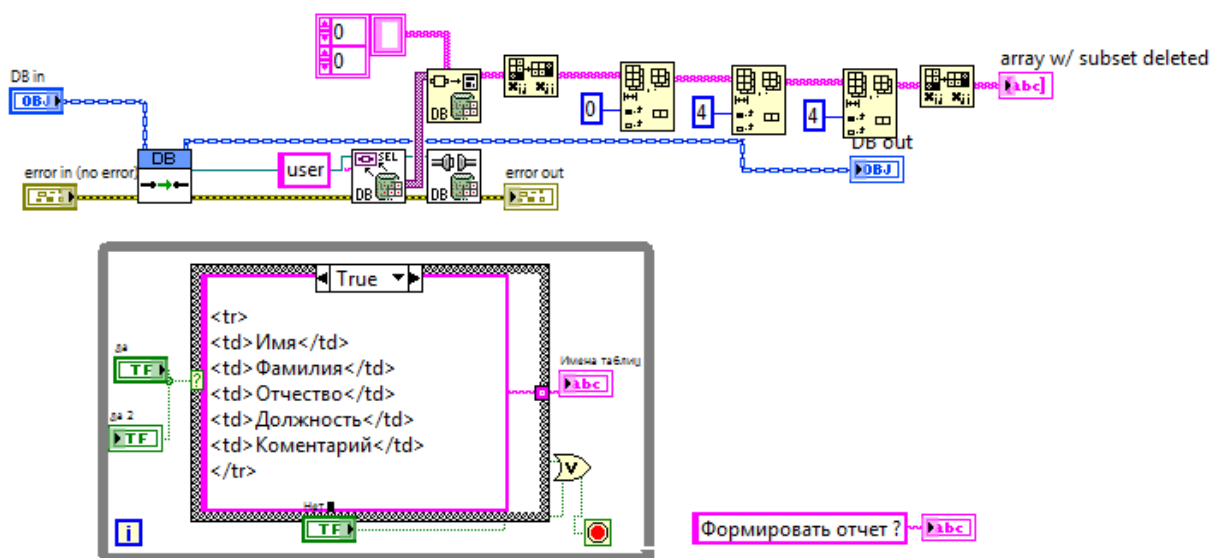


Рисунок 34 – Законченная функция по чтению данных по пользователю

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок 35).



Рисунок 35 – Иконка функции для чтения данных по пользователю

#### 4.3.2 Чтение записей по тиристорам из базы данных

Чтение из базы данных производится в несколько этапов. Первым этапом служит подключение к базе данных. Далее для работы с базой данных мы берем функцию для записи информации «DB Tools Select Data».

На вход «connection information» мы должны провести линию связи с функцией «DB Tools Open Connec», на вход «table» мы подаем название таблицы в базе данных с которой мы хотим работать в этом случае это «Thyristor», если мы хотим вывести определенные столбцы таблицы то на вход «columns» мы должны подать в формате массив те столбцы которые мы хотим считать с базы данных и последним обязательным пунктом является проведение линии ошибки на вход «error in». В итоге мы получили такую схему подключения (рисунок 36).

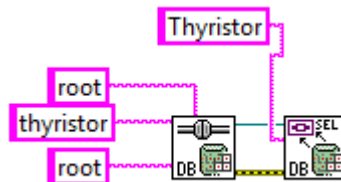


Рисунок 36 – Схема подключения функции DB Tools Select Data

После запроса в базу данных на вывод таблицы «Thyristor» нужно ее вывести. Для этого мы используем функцию «Database Variant To Data».

На вход «database variant» мы должны провести линию связи с выхода «date» функции «DB Tools Select Data», на вход «type» мы подаем тип, в котором нужно выводить информацию, в данном случае это массив, на выход «date» мы ставим индикатор. Завершающим этапом является

отключение от базы с помощью функции «DB Tools Clous Connection VI». В итоге мы получили такую схему подключения (рисунок 37).

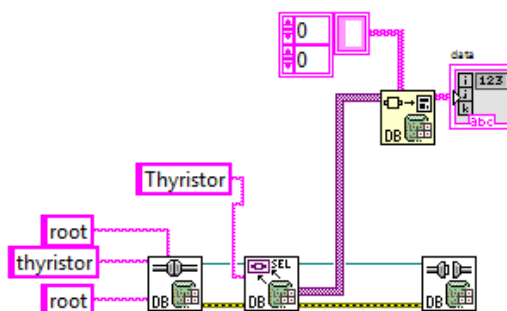


Рисунок 37 – Полная схема для чтения записям по тиристорам из базы

После завершения схемы для чтения записей по тиристорам, необходимо проверить её работоспособность. В базе данных было сделано две записи для того, чтобы было наглядно видно, как работает схема (рисунок 38).

```
1 • SELECT * FROM mydb.thyristor;
```

Id	The_name_Thyristor	Manufacturer	Year_of_production	Passport_Ugt	Passport_Igt	Comment
1	T122-25-1	NXP Semiconductors	2015	60	2.5	comment 1
2	T142-50-13	Unisonic Technologies	2011	120	3	comment 2
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 38 – Записи в базе данных

После запуска программы на индикатор выводиться информация, идентичная информации, которая у нас записана в базе данных, это значит что функция по чтению измерений работает исправно (рисунок 39).

data							
0	1	T122-25-1	NXP	2015	60	2,5	comment 1
0	2	T142-50-13	Unisonic	2011	120	3	comment 2

Рисунок 39 – Индикатор для вывода информации с функции Database Variant To Data

Перед использованием в основной программе программа была переделана и сделана подклассом. (рисунок 40).

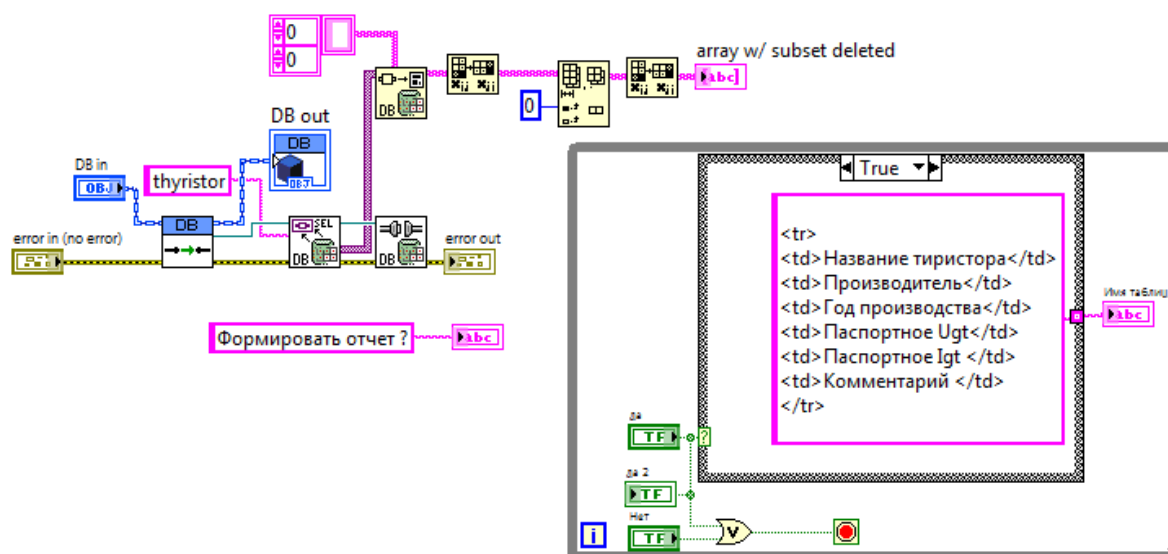


Рисунок 40 – Законченная функция по чтению данных по тиристор

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок 41).



Рисунок 41 – Иконка функции для чтения данных по тиристор

### 4.3.3 Чтение записей по измерениям из базы данных

Чтение из базы данных производится в несколько этапах. Первым этапом служит подключение к базе данных. Далее для работы с базой данных мы берем функцию для записи информации «DB Tools Select Data».



На вход «connection information» мы должны провести линию связи с функцией «DB Tools Open Connec», на вход «table» мы подаем название таблицы в базе данных с которой мы хотим работать в этом случае это «Measurement», если мы хотим вывести определенные столбцы таблицы то на вход «columns» мы должны подать в формате массив те столбцы которые мы хотим считать с базы данных и последним обязательным пунктом является проведение линии ошибки на вход «error in». В итоге мы получили такую схему подключения (рисунок 42).

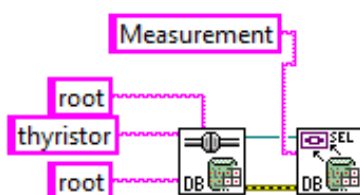


Рисунок 42 – Схема подключения функции DB Tools Select Data

После запроса в базу данных на вывод таблицы «Measurment» нужно ее вывести. Для этого мы используем функцию «Database Variant To Data».

На вход «database variant» мы должны провести линию связи с выхода «date» функции «DB Tools Select Data», на вход «type» мы подаем тип, в котором нужно выводить информацию, в данном случае это массив, на выход «date» мы ставим индикатор. Завершающим этапом является отключение от базы с помощью функции «DB Tools Close Connection VI». В итоге мы получили такую схему подключения (рисунок 43).

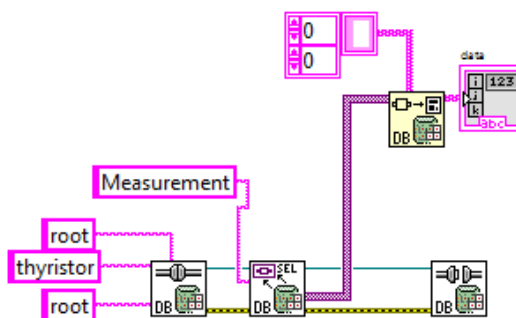


Рисунок 43 – Полная схема для чтения записей по измерениям из базы

После завершения схемы для чтения записей по измерениям, необходимо проверить её работоспособность. В базе данных было сделано три записи для того, чтобы было наглядно видно, как работает схема (рисунок 44).

The screenshot shows a SQL query window with the following SQL statement: `SELECT * FROM mydb.Measurement;`. Below the query, a 'Result Grid' displays the following data:

Id	Thyristor_Id	User_Id	Igt	Ugt	Date	Time	Comment
1	2	2	0.9	20	2017-05-14	15:16:17	NULL
2	1	1	1.2	10	2018-04-15	16:14:27	NULL
3	1	2	1.3	30	2018-05-21	16:18:20	coment1

Рисунок 44 – Записи в базе данных

После запуска программы на индикатор выводиться информация, идентичная информации, которая у нас записана в базе данных, это значит, что функция по чтению измерений работает исправно (рисунок 45).

The screenshot shows a data indicator window titled 'data' with a table containing the following information:

1	2	2	0,9	20	14.05.2017	05.06.2018 15:16:	
2	1	1	1,2	10	15.04.2018	05.06.2018 16:14:	
3	1	2	1,3	30	21.05.2018	05.06.2018 16:18:	coment1

Рисунок 45 – Индикатор для вывода информации с функции Database Variant To Data

Перед использованием в основной программе программа была переделана и сделана подклассом. Нижний цикл сделан для формирования отчета (рисунок 46).

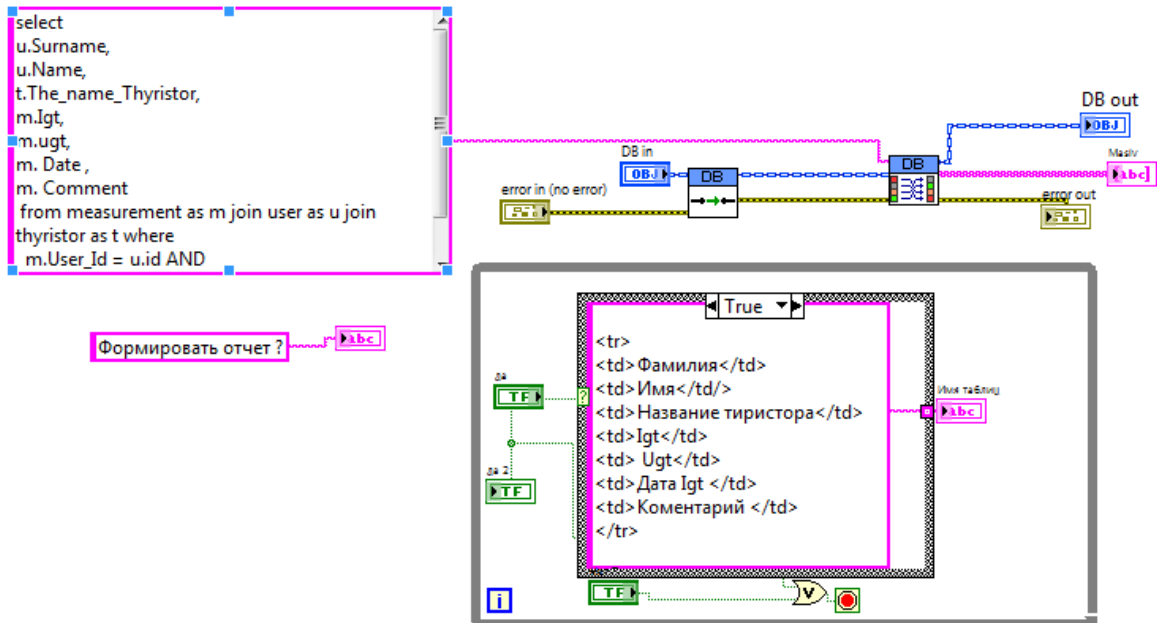


Рисунок 46 – Законченная функция по чтению данных по измерению

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок 47).

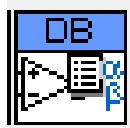


Рисунок 47 – Иконка функции для чтения данных по измерению

#### 4.4 Разработка инструмента для формирования запроса в базу

Создание запроса в базу данных заключается в несколько этапов. Первым этапом является подключение к базе данных. Далее для того, чтобы сделать запрос мы используем функцию «DB Tools Execute Query VI».

На вход «connection information» мы должны провести линию связи с функцией «DB Tools Open Connec», на вход «SQL query» мы подаем определенную команду в данном случае мы подаем пустую константу, на вход «cursor type» ставим «static» для того, чтобы у нас был статичный курсор и последним обязательным пунктом является проведение линии ошибки на вход «error in». Остальные входы не играют важную роль в работе с базой данных. В итоге мы получили такую схему подключения (рисунок 48).

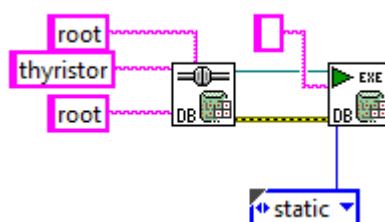


Рисунок 48 – Схема подключения функции DB Tools Execute Query VI

Дальше для того, чтобы вывести информацию получить информацию по нашему запросу мы используем функцию «DB Tools Fetch Recordset Data VI». Потом для того, чтобы отобразить информацию мы подключаем к выходу «recordset data» функцию «Database Variant To Data».

На вход «type» мы подаем тип, в котором нужно выводить информацию, в данном случае это массив, на выход «date» мы ставим индикатор и последним обязательным пунктом является проведение линии ошибки на вход «error in». Остальные входы не играют важную роль в работе с базой данных. В итоге мы получили такую схему подключения (рисунок 49).

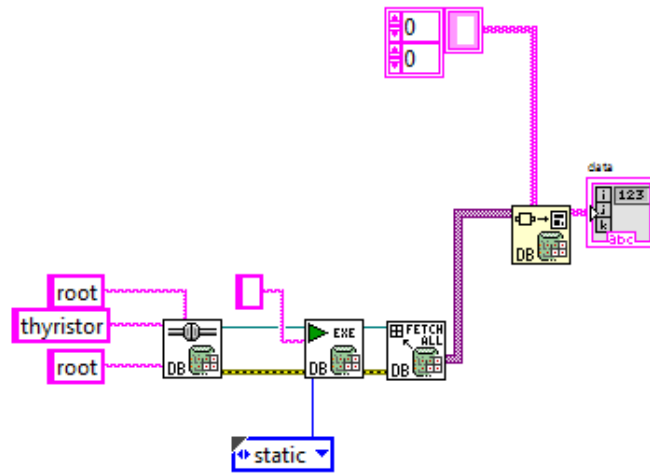


Рисунок 49 – Схема подключения функций DB Tools Fetch Recordset Data VI и Database Variant To Data

Потом для очистки запросов в базу данных мы используем функцию «DB Tools Free Object VI» а для закрытия связи с базой данных мы используем функцию «DB Tools Close Connection VI» (рисунок 50).

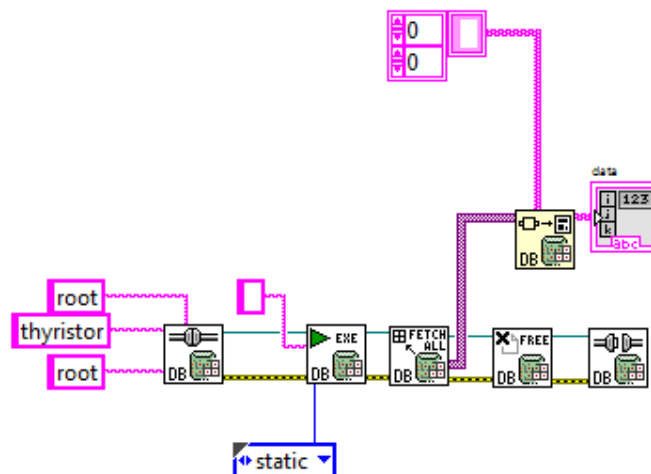


Рисунок 50 – Полная схема для формирования запросов в базу данных

Функция для работы с базой данных закончена. Для проверки выведем столбцы из разных таблиц базы данных.

Выведем название всех тиристоров, которые записаны в базе данных имена и фамилии пользователей, которые проводили с ними измерения, результаты их измерений и время измерения. Для этого мы используем такой запрос:

```

select
t.The_name_Thyristor,
u.Surname,
u.Name,
m.Igt,
m.ugt,
m.Time
from measurement as m join user as u join thyristor as t where
m.User_Id = u.id AND
m.Thyristor_Id = t.id

```

После ввода в базу данных этого запроса получил результат (рисунок 51).

The screenshot shows a SQL query editor with the following text:

```

1 select
2 t.The_name_Thyristor,
3 u.Surname,
4 u.Name,
5 m.Igt,
6 m.ugt,
7 m.Time
8 from measurement as m join user as u join thyristor as t where
9 m.User_Id = u.id AND
10 m.Thyristor_Id = t.id

```

Below the query editor is a 'Result Grid' window displaying the following data:

The_name_Thyristor	Surname	Name	Igt	ugt	Time
T142-50-13	Kamarov	Sergey	0.9	20	15:16:17
T122-25-1	Sarov	Nikolay	1.2	10	16:14:27
T122-25-1	Kamarov	Sergey	1.3	30	16:18:20

Рисунок 51 – Записи по запросу

Теперь введем этот запрос в функцию «DB Tools Execute Query VI» (рисунок 52).

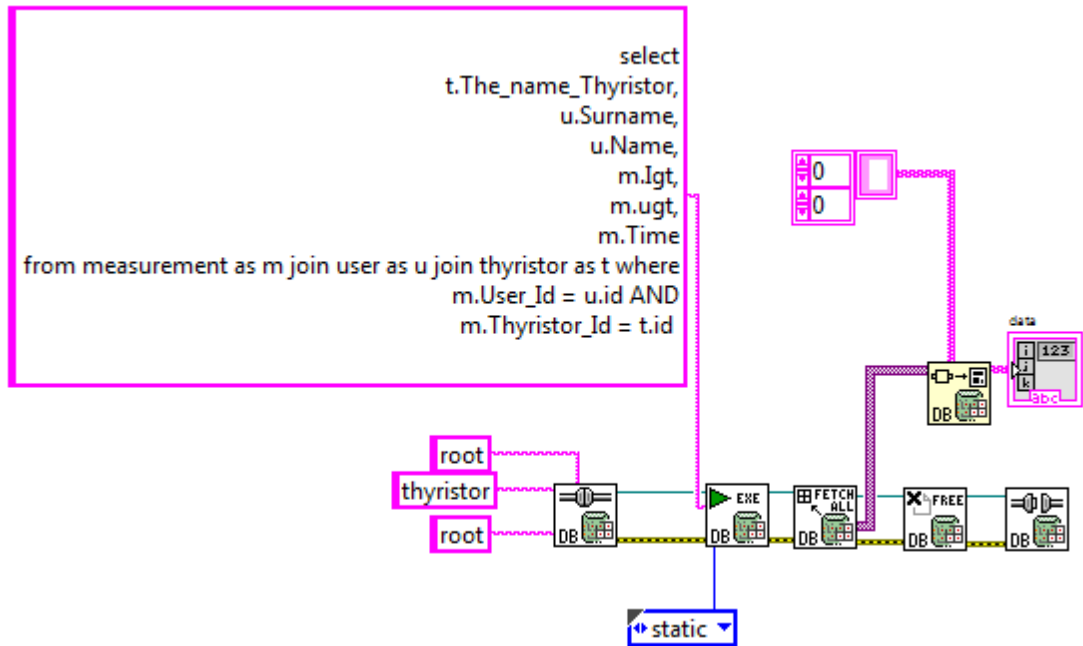


Рисунок 52 – Записи по запросу

По запросу на индикатор вышла информация идентичная что и по запросу в базе данных, а это значит, что функция исправно работает (рисунок 53).

data						
0	T142-50-13	Kamarov	Sergey	0,9	20	05.06.2018 15:16:
0	T122-25-1	Sarov	Nikolay	1,2	10	05.06.2018 16:14:
	T122-25-1	Kamarov	Sergey	1,3	30	05.06.2018 16:18:

Рисунок 53 – Индикатор для вывода информации с функции Database Variant To Data

Перед использованием в основной программе программа была переделана и сделана подклассом (рисунок 54).

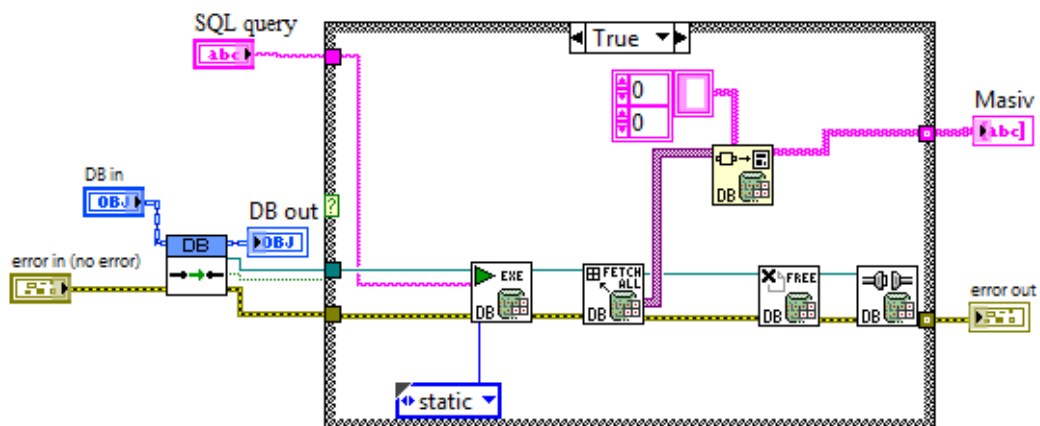


Рисунок 54 – Законченная функция для обращения в базу

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок 55).



Рисунок 55 – Иконка функции для чтения данных по измерению



## 5 Разработка инструментов для формирования отчета

Формирование отчета является довольно важной частью работы пользователя. Формироваться отчет будет выводиться на странице браузера из-за того, что такая форма вывода отчета является наиболее проста в понимании для пользователя и наглядна.

В программируемой среде LabVIEW был разработан инструмент для формирования программного кода отчетной страницы в браузере компьютера. Инструмент берет массив из базы данных разделяет информацию на столбцы и строки и приписывает теги. После формирования кода она создает файл и загружает полученный скетч в него и потом открывает созданный файл с загруженным скетчем на странице браузера (рисунок 56).

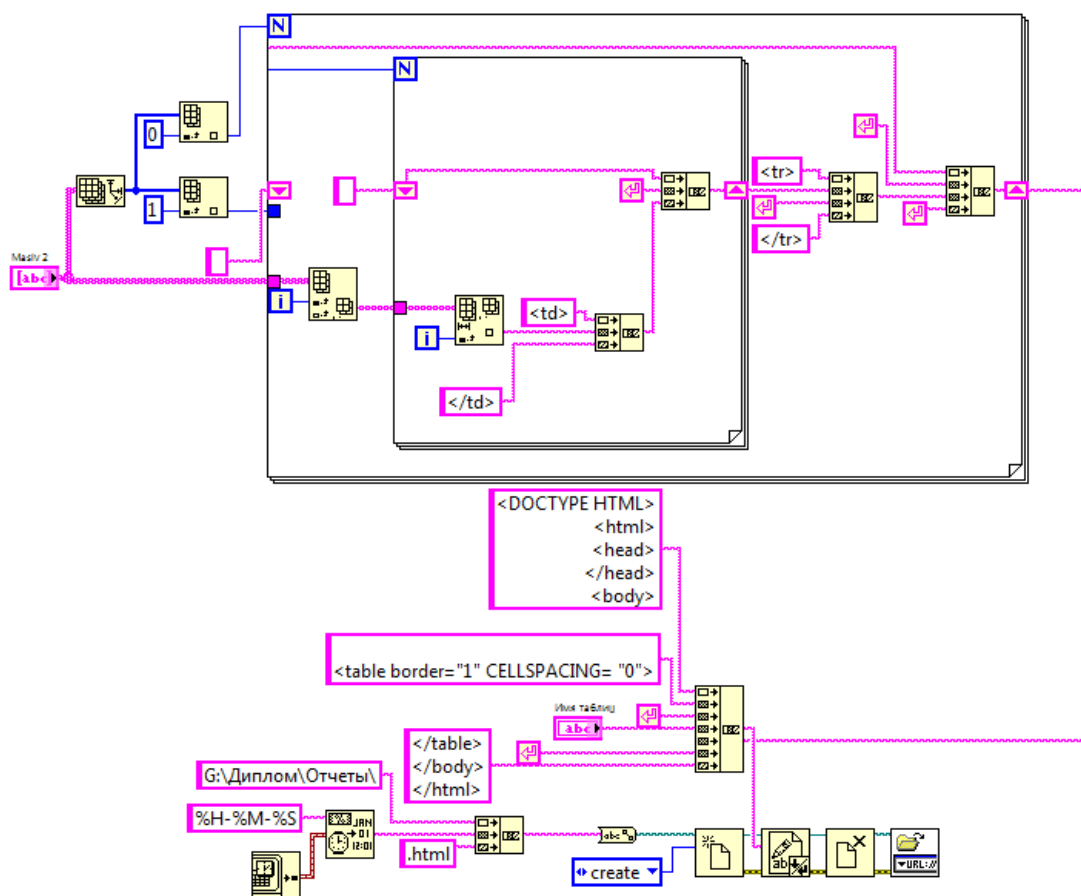


Рисунок 56 – Инструмент для формирования отчета

Изм.	Лист	№ докум.	Подпись	Дата

Для проверки работоспособности инструмента возьмем массив по пользователям из базы данных (рисунок 57).

array w/ subset deleted					
0	Nikolay	Sarov	Nikolayevich	Student	comment 1
0	Sergey	Kamarov	Sergeevich	Teacher	comment 2

Рисунок 57 – Массив по пользователю

Потом подадим его на вход инструмента. Он разбивает массив на строки и столбы и вставляет скетчи формирования таблицы из этого получается программный код для формирования отчета на странице браузера:

```
<DOCTYPE HTML>
<html>
  <head>
  </head>
<body>
  <table border="1" CELLSPACING="0">
    <tr>
      <td>Имя</td>
      <td>Фамилия</td>
      <td>Отчество</td>
      <td>Должность</td>
      <td>Комментарий</td>
    </tr>
    <tr>
      <td>Nikolay</td>
      <td>Sarov</td>
      <td>Nikolayevich</td>
      <td>Student</td>
      <td>comment 1</td>
    </tr>
  </table>

```

```

</tr>
<tr>
    <td>Sergey</td>
    <td>Kamarov</td>
    <td>Sergeevich</td>
    <td>Teacher</td>
    <td>comment 2</td>
</tr>
</table>
</body>
</html>

```

После формирования программного кода он записывается в созданный файл с форматом "html" и открывается в следствии этого выводится отчет на страницу браузера (рисунок 58).

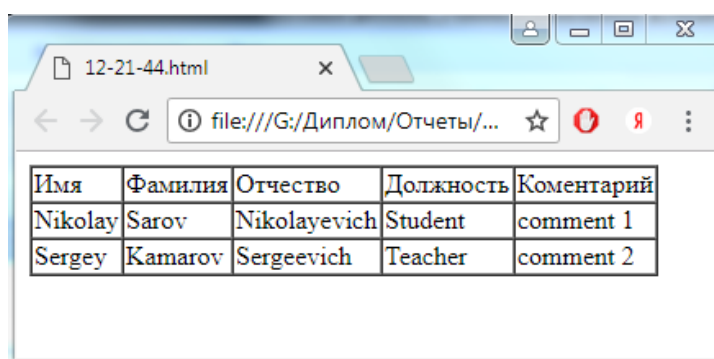


Рисунок 58 – Сформированный отчет на странице браузера

Для уменьшения массива программного кода в дальнейшем эта программа будет использоваться как функция и отображаться иконкой (рисунок 59).



Рисунок 59 – Иконка функции для создания отчета

## 6 Разработка программного кода для управления аппаратной частью испытательного устройства

### 6.1 Инструмент для выполнения измерений

Для начала работы с измерительным устройством нужно инициализировать его входы. В функциях для инициализации были указаны входы, и метод измерения, так же установлены пределы измерения. Кейс введен для возникновения ошибки (рисунок 60).

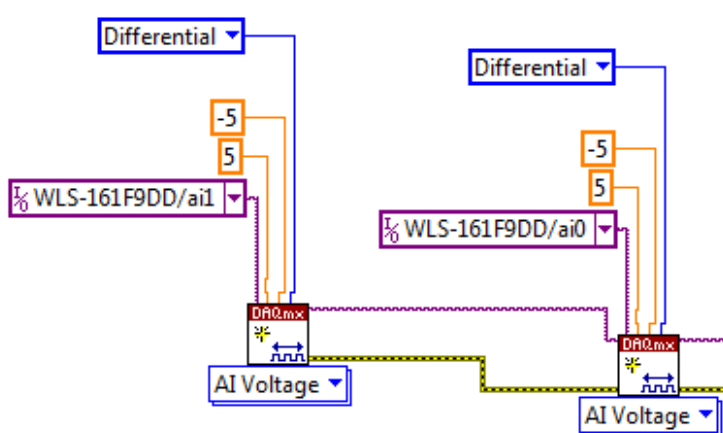


Рисунок 60 – Инициализация

После получение информации с входов мы проводим необходимые действия для получения измерений (рисунок 61).

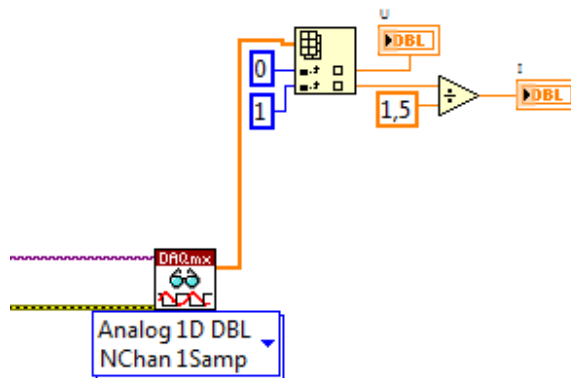


Рисунок 61 – Вычисление

После того как проведены необходимые действия для получения измерений, нужно очистить каналы для следующих измерений. Для этого было реализовано с помощью функций очищение каналов программный код (рисунок 62).

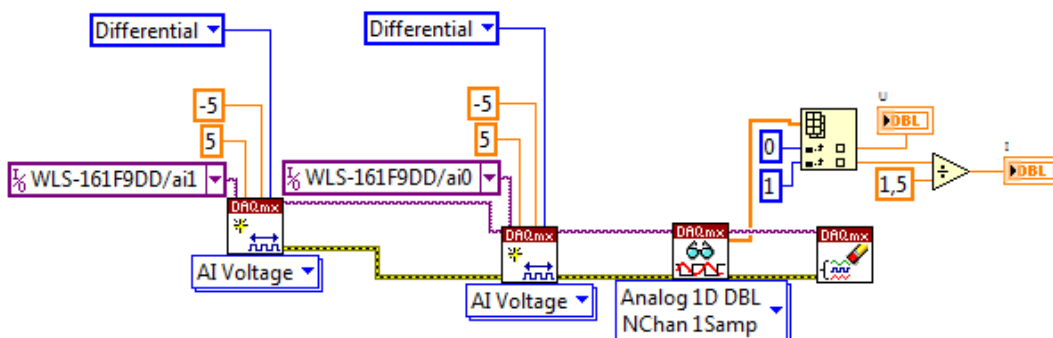


Рисунок 62 – Инструмент для выполнения измерения

## 6.2 Программный код контроллера

Создание целостного прибора имеющий связь с ПК заключается в двух этапах, создание аппаратной части и программной части прибора. Но для того, чтобы эти две части функционировали между ними должен быть налажен обмен данными.

В приборе для нахождения параметров силового тиристора связь между аппаратной и программной частью будет налажена с помощью программируемого контроллера Arduino Uno с подключенным к нему Ethernet shield. С помощью Ethernet shield мы создаем webserver для общения с контроллером.

Для работы был разработан алгоритм (рисунок 63).

					<b>БР-02069964-12.03.01-10-18</b>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		50

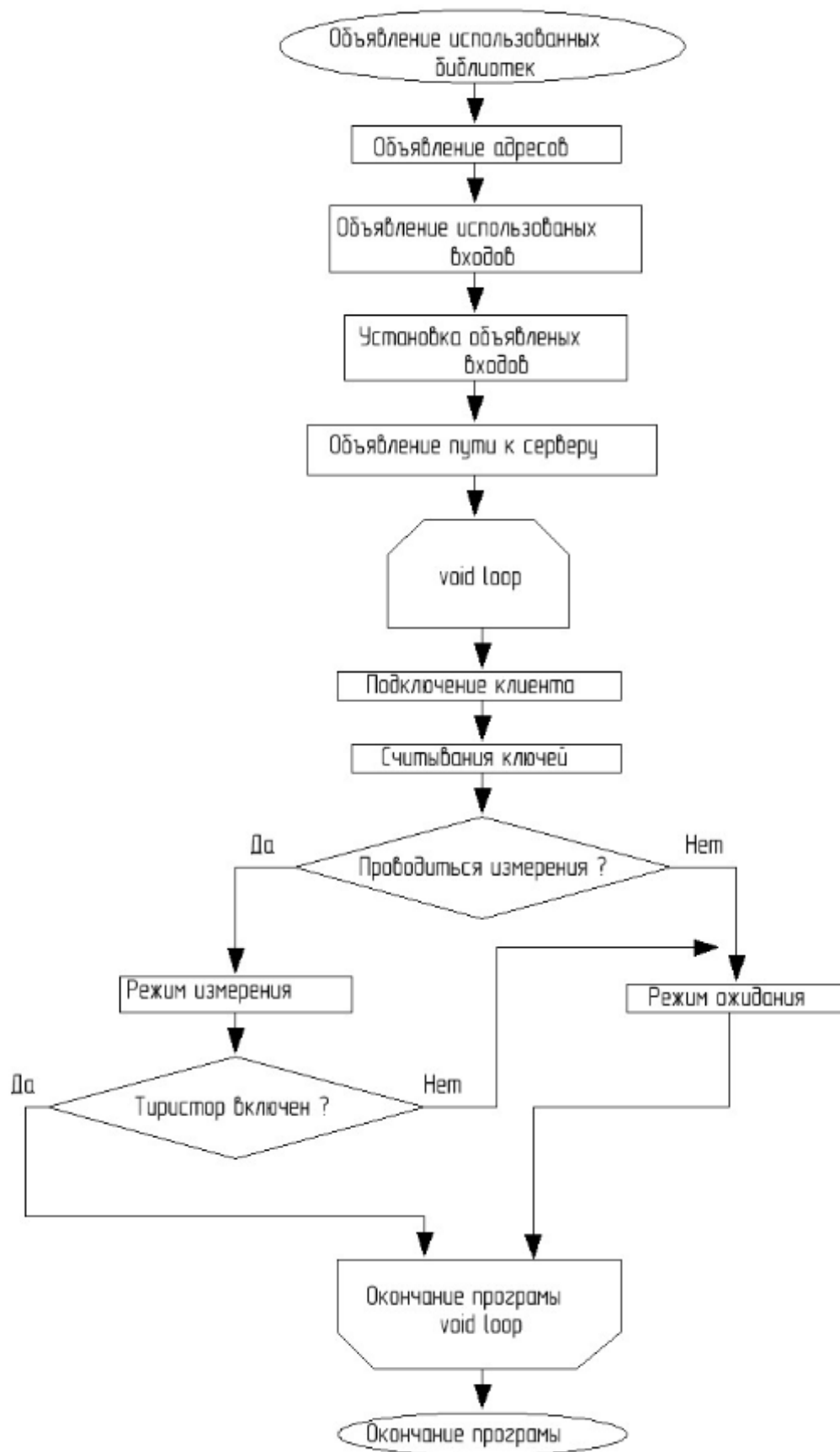


Рисунок 63 – Алгоритм программирования Arduino

В начале создания программы для управления аппаратной частью нужно прописать библиотеки, так как некоторым элементам они требуются для правильной работы. Объявленные библиотеки:

```
//Библиотеки SPI
#include <SPI.h>
// Библиотеки Ethernet
#include <Ethernet.h>
```

Следующим шагом является введение списка используемых переменных и соотнесение их с используемыми ножками, так же объявление IP и MAC адресов использованных устройств. Объявление переменных:

```
// MAC адрес Ethernet Shield
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
// IP адрес будущего сервера
IPAddress ip(192, 168, 1, 10);
// IP адрес клиента
IPAddress IOCPServer(192, 168, 1, 1);
boolean isConnect = 0;
EthernetClient client;

// Соотнесение переменных с ножками
int led = 2;
int led1 = 4;
int led2 = 5;
int digatalPin = 3;
```

Следующим является цикл «void setup». В этом цикле прописаны команды, которые используются всего один раз. Например, сразу после загрузки контроллера. Здесь прописан запуск последовательного порта с



необходимой скоростью, установка элементов на вход или выход, условие запуска, инициализация датчика, выделение памяти. Прописывание «void setup»:

```
void setup()
{
// Запуск последовательного порта
  Serial.begin(9600);
// Объявление задержки
  delay(100);
  while (!Serial) {
  }
// установка контактов в режим вывода и ввода
  pinMode(digitalPin, INPUT);
  pinMode(led, OUTPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
// установка 4-го контакта в активный режим
  digitalWrite(led1, HIGH );
}
//Объявление используемых источников
  boolean EthConnect(){
  Ethernet.begin(mac, ip);
// Объявление задержки
  delay(10);
// Объявление строки приподключении
  Serial.println("connecting...");

  // Объявление пути
  if (client.connect(IOCPServer, 8090)) {
// Объявление строки после подключения
  Serial.println("connected");
  //isConect = 1;
  return true;
  }
```

```

// Объявление условий при не подключении
else {
// Объявление строки при не подключении
Serial.println("connection failed");
//isConect = 0;
return false;
}
}

```

Цикл «void setup» используется для инициализации, которые крутятся в цикле. Позволяет совершать вычисления и реагировать на них. В этом цикле будут объявлены ключи для управления аппаратной частью прибора.

Прописывание «void setup»:

```

void loop(){
// Упрвление работой прибора
while (!isConnect){
isConnect = EthConnect();
delay(500);
}
if (!client.connected()) {
Serial.println();
Serial.println("disconnecting");
client.stop();
isConnect = false;
}
delay (1000);
if (digitalRead(digatalPin)== LOW){
client.print("1");
}
else {
client.print("2");
} }
if (isConnect){
if (client.available() > 0)
{

```

```

delay (10);
char c = client.read();
Serial.println(c);
delay (10);
// Объявление ключей
switch ( c )
{
// Объявление ключа к подготовке прибора к измерению
case 'm':
    Serial.println("start measurement");
    //client.print("m_OK");
    digitalWrite(led, HIGH); // вывод №2 в активное состояние
    digitalWrite(led1, LOW ); // вывод №4 в активное состояние
    digitalWrite(led2, HIGH); // вывод №5 в активное состояние
    delay(1000); // пауза 1-секунда
    digitalWrite(led1, HIGH );
    digitalWrite(led, LOW); // вывод №2 в неактивное состояние
    delay(1000); // пауза 1-секунда
    break;
    // Ключ к переводу прибора в режим ожидания
case 's':
    digitalWrite(led, LOW); // вывод №2 в неактивное
    digitalWrite(led1, HIGH); // вывод №4 в неактивное
    digitalWrite(led2, LOW); // вывод №5 в неактивное
    break;
}
}
}
}
}

```

После формирования кода на программируемый контроллер нужно реализовать связь между программой и контроллерами. Это делается с помощью функций общения ТСР (рисунок 64). С помощью этого инструмента будет посылаться название ключей в webserver в следствии чего контроллер будет выполнять прописанные в этом ключе команды.

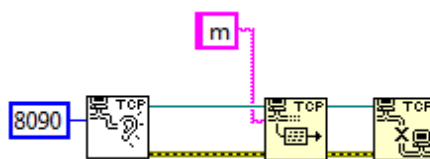


Рисунок 64 – Функция для записывания в сервер ключей

Так же был сделан инструмент для получения данных из web server для того что бы следить открылся ли тиристор (рисунок 65).

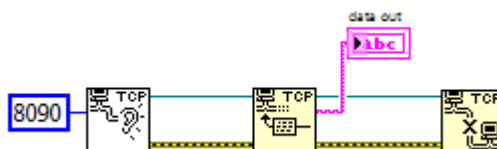


Рисунок 65 – Функция для считывания с сервера информации

## ЗАКЛЮЧЕНИЕ

Работа с измерительным прибором является довольно важной частью работы инженера. Обеспечивание связи между компьютером и прибором намного упрощает работу с измерениями и открывает возможность работы с базами данных. Базы данных в свою очередь является довольно эффективной средой по хранению и обработке информации. Все эти достоинства и включает в себя программное обеспечение любого прибора.

					<b>БР-02069964-12.03.01-10-18</b>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		57

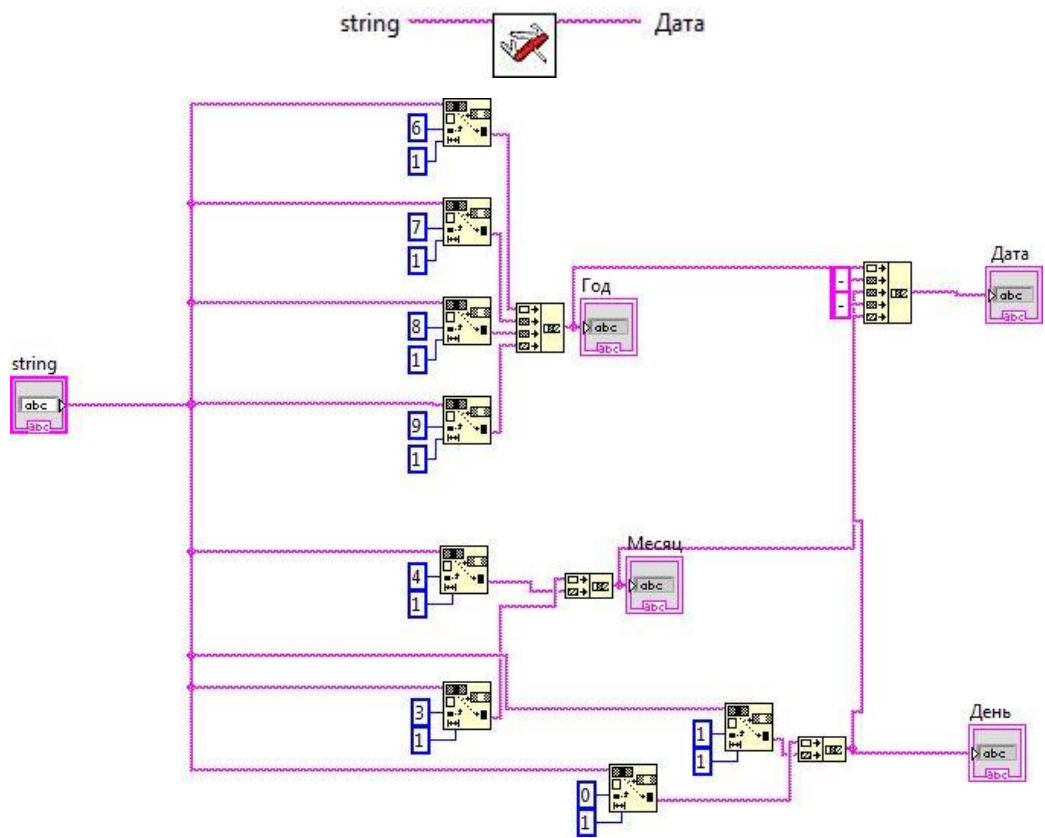
## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Стюарт Болл Р. Аналоговые интерфейсы микроконтроллеров / Стюарт Болл Р. – Москва: Издательский дом «Додэка-XXI», 2007. – 21 с.
2. Программирование Ардуино [Электронный ресурс] // [www.arduino.ru](http://www.arduino.ru). Режим доступа: <http://arduino.ru/Reference> (дата обращения: 10.09.2017)
3. Культин Н. Б. Основы программирования в Delphi XE / Н. Б. Культин. — СПб.: БХВ-Петербург, 2011. — 416 с.
4. Бутырин П.А. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7 / П.А. Бутырин, Т.А. Васьковская, В.В. Каратаева, С.В. Материкин. — М.: ДМК Пресс, 2005. — 266 с.
5. Митин И.В. Анализ и обработка экспериментальных данных / И.В. Митин, В.С. Русаков. — М.: Физич.ф-т. МГУ, 2002. — 126 с.
6. Яргер Р. Дж. MySQL и mSQL: Базы данных для небольших предприятий и Интернета / Р. Дж. Яргер, Дж. Риз, Т. Кинг. - М.: СПб: Символ-Плюс, 2014. - 560 с.
7. Дубнов П.Ю. Access 2000. Проектирование баз данных / П.Ю. Дубнов. - М.: ДМК, 2000. - 272 с.
8. Ковязин, А.Н. Архитектура, администрирование и разработка приложений баз данных в InterBase/FireBird/Yaffil / А.Н. Ковязин, С.М. Востриков. - М.: Кудиц-образ; Издание 4-е, 2006. - 496 с.

									Лист
									58
Изм.	Лист	№ докум.	Подпись	Дата	БР-02069964-12.03.01-10-18				

**ПРИЛОЖЕНИЕ А**  
**(обязательное)**  
**программный код**

					<b>БР-02069964-12.03.01-10-18</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		59



Изм.	Лист	№ докум.	Подпись	Дата

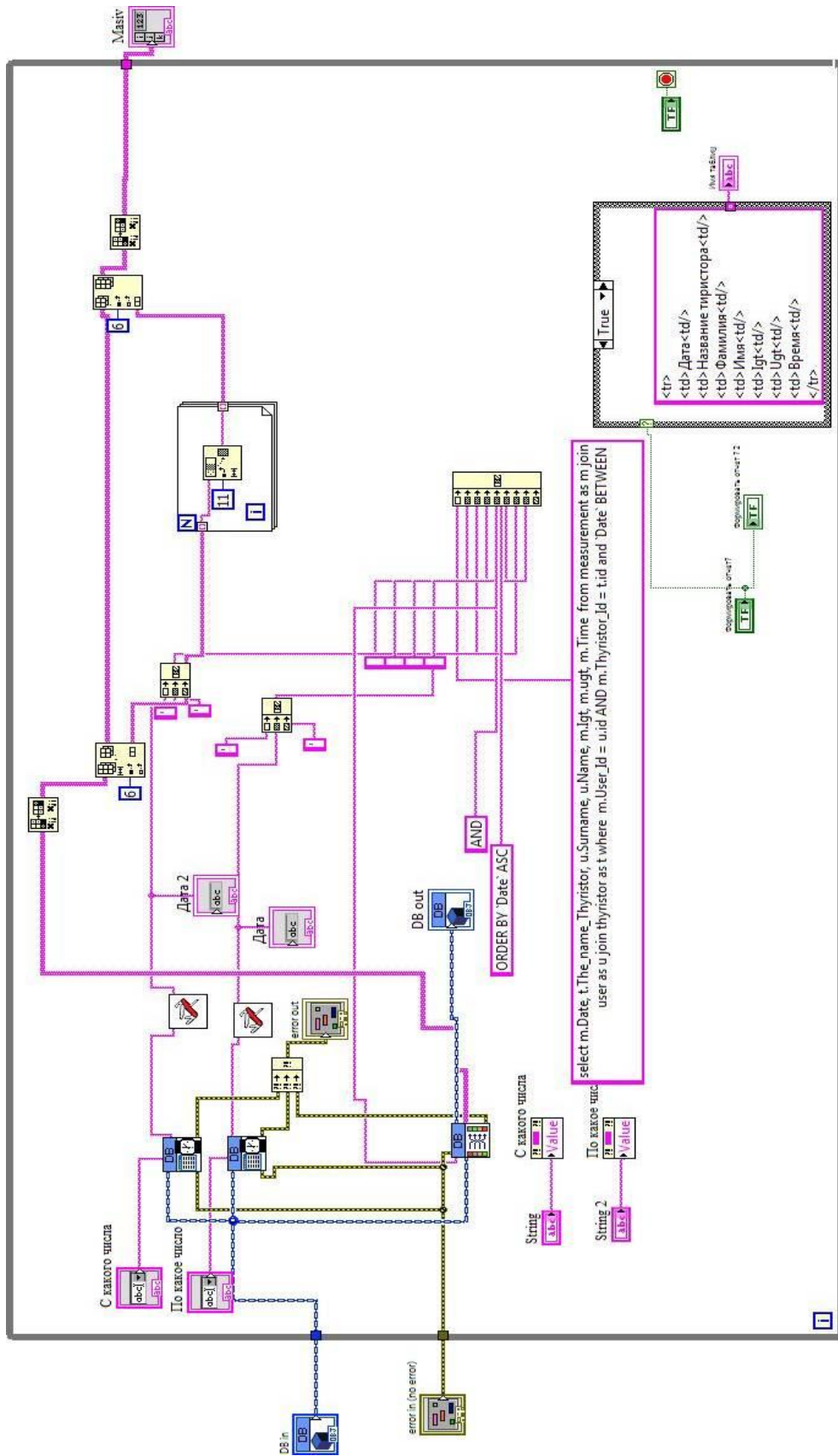
БР-02069964-12.03.01-10-18

Лист

60



По какое число  
 С какого числа  
 DB in  
 error in (no error)  
 Masiv  
 DB out  
 Имя таблиц  
 error out  
 Формировать отчет ? 2

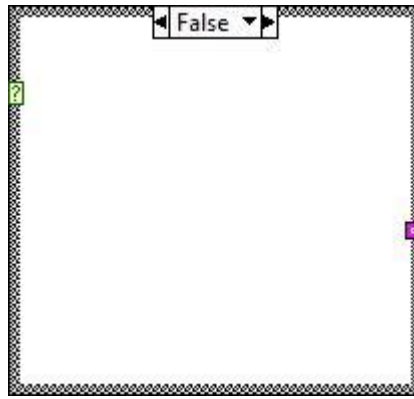


Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

Лист


61




С какого числа

По какое число

Формировать отчет?

 отчет

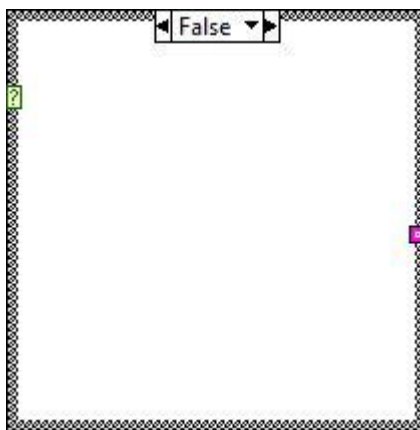
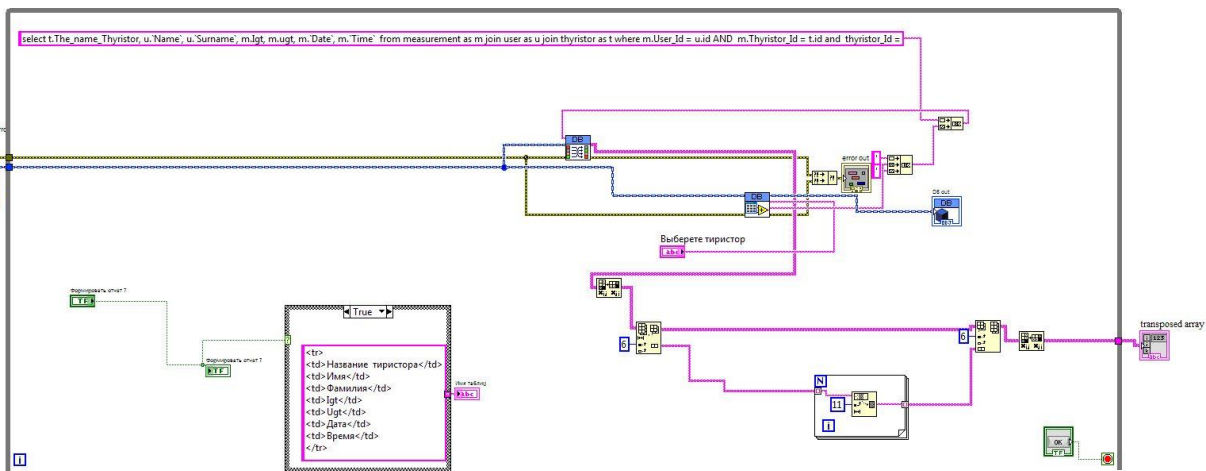
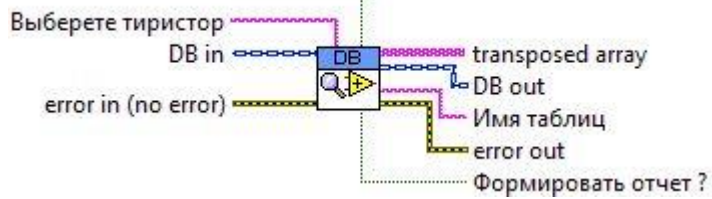
 поиск

Изм.	Лист	№ докум.	Подпись	Дата

**БР-02069964-12.03.01-10-18**

Лист

62



Выберете тиристор

Формировать отчет ?

Отчет

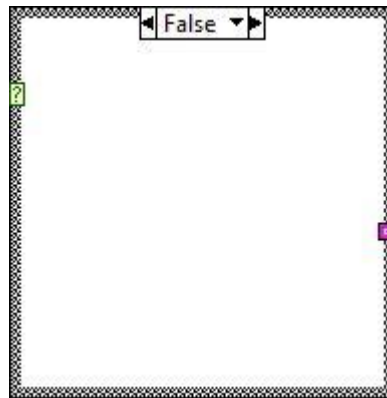
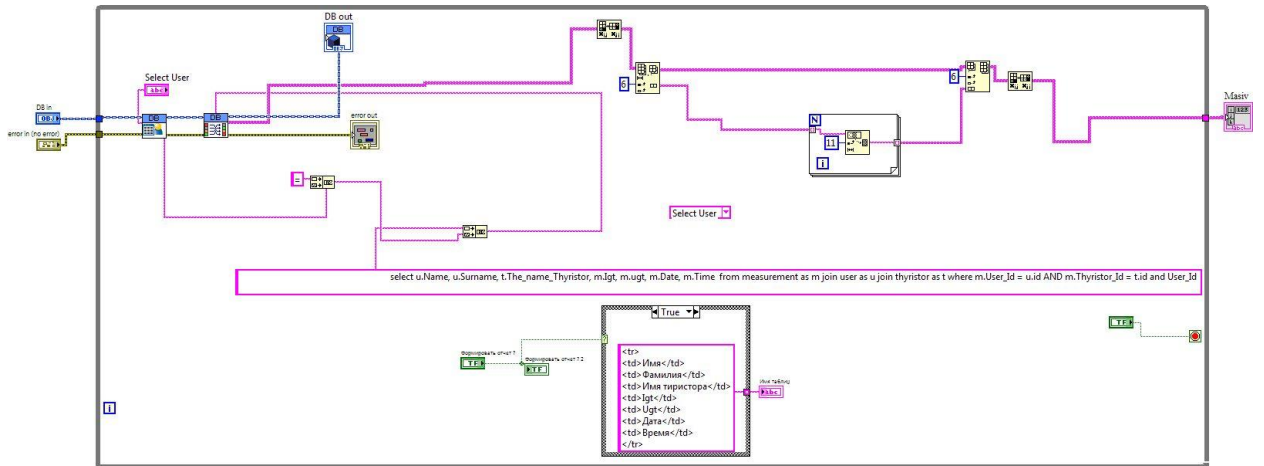
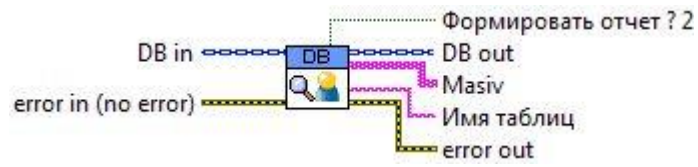
Поиск

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

Лист

63



Выберите пользователя для поиска

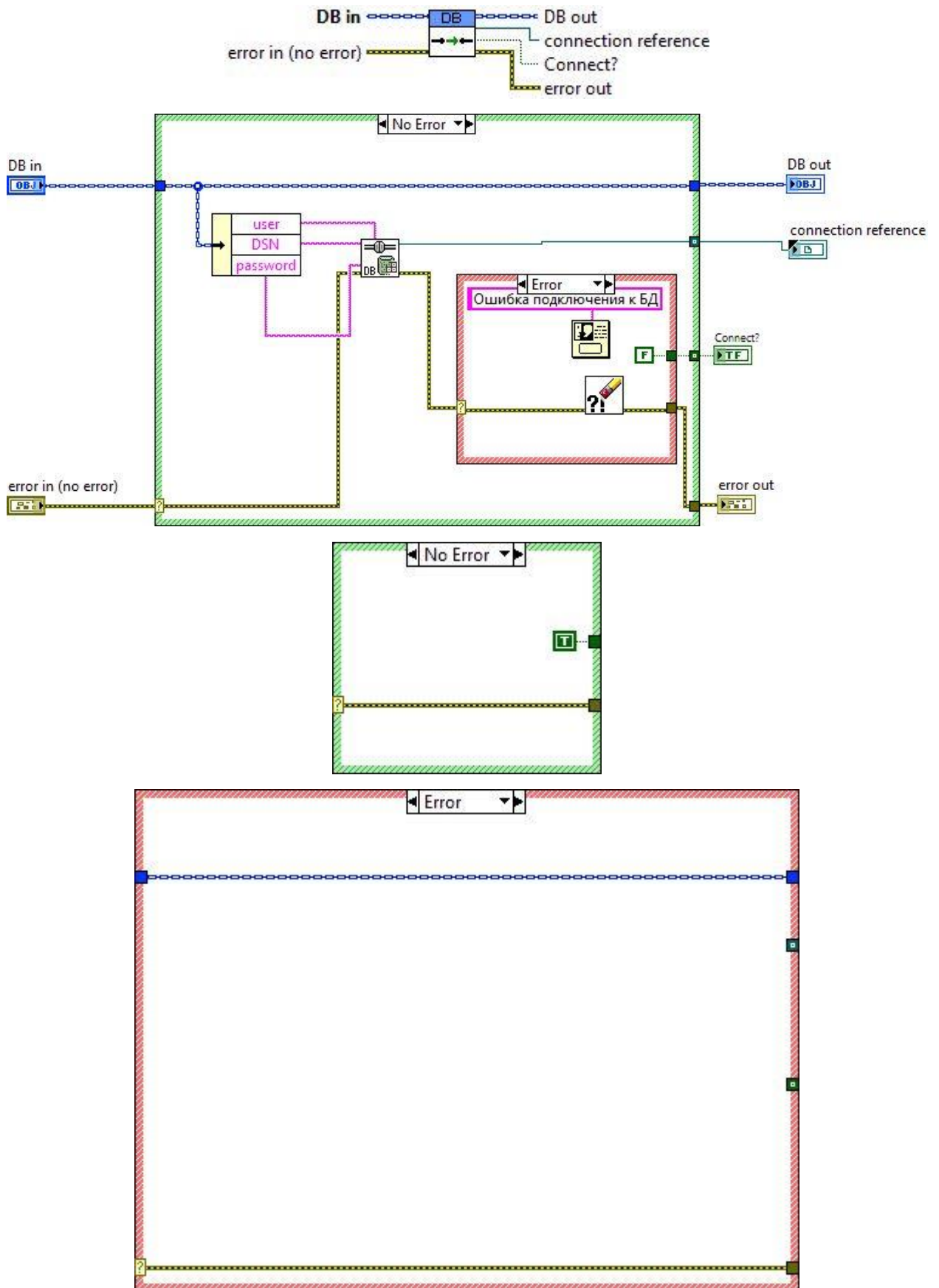
Формировать отчет ?

Отчет

Поиск

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

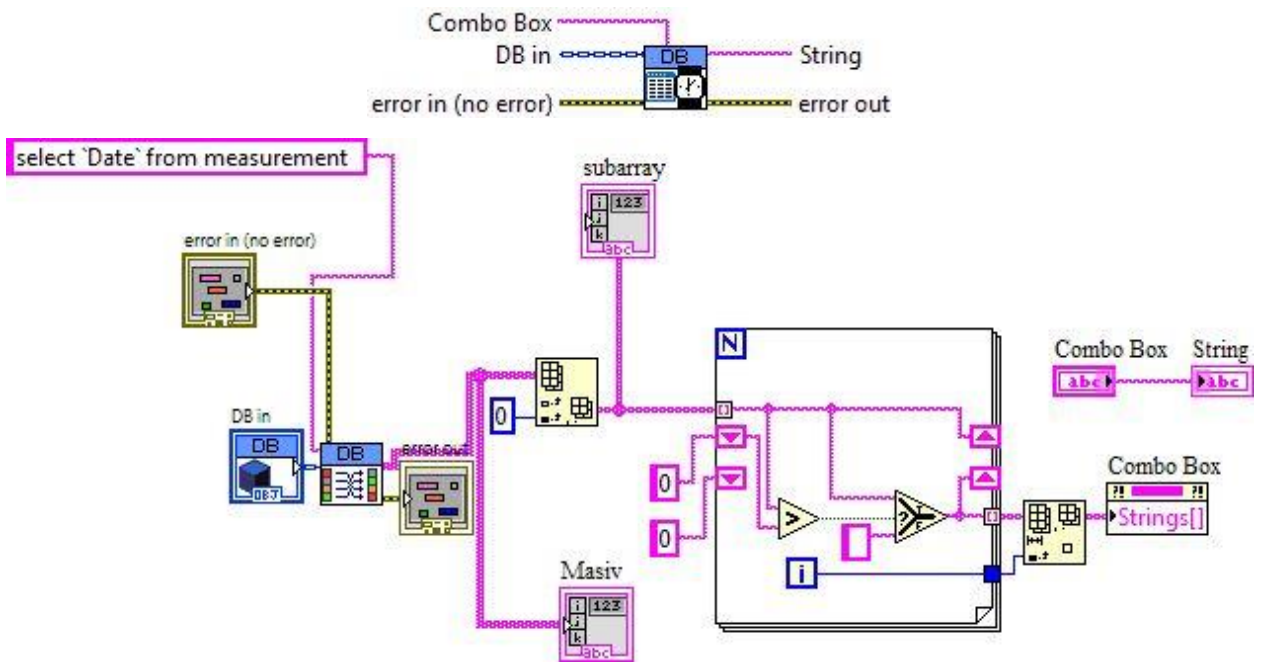


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

65

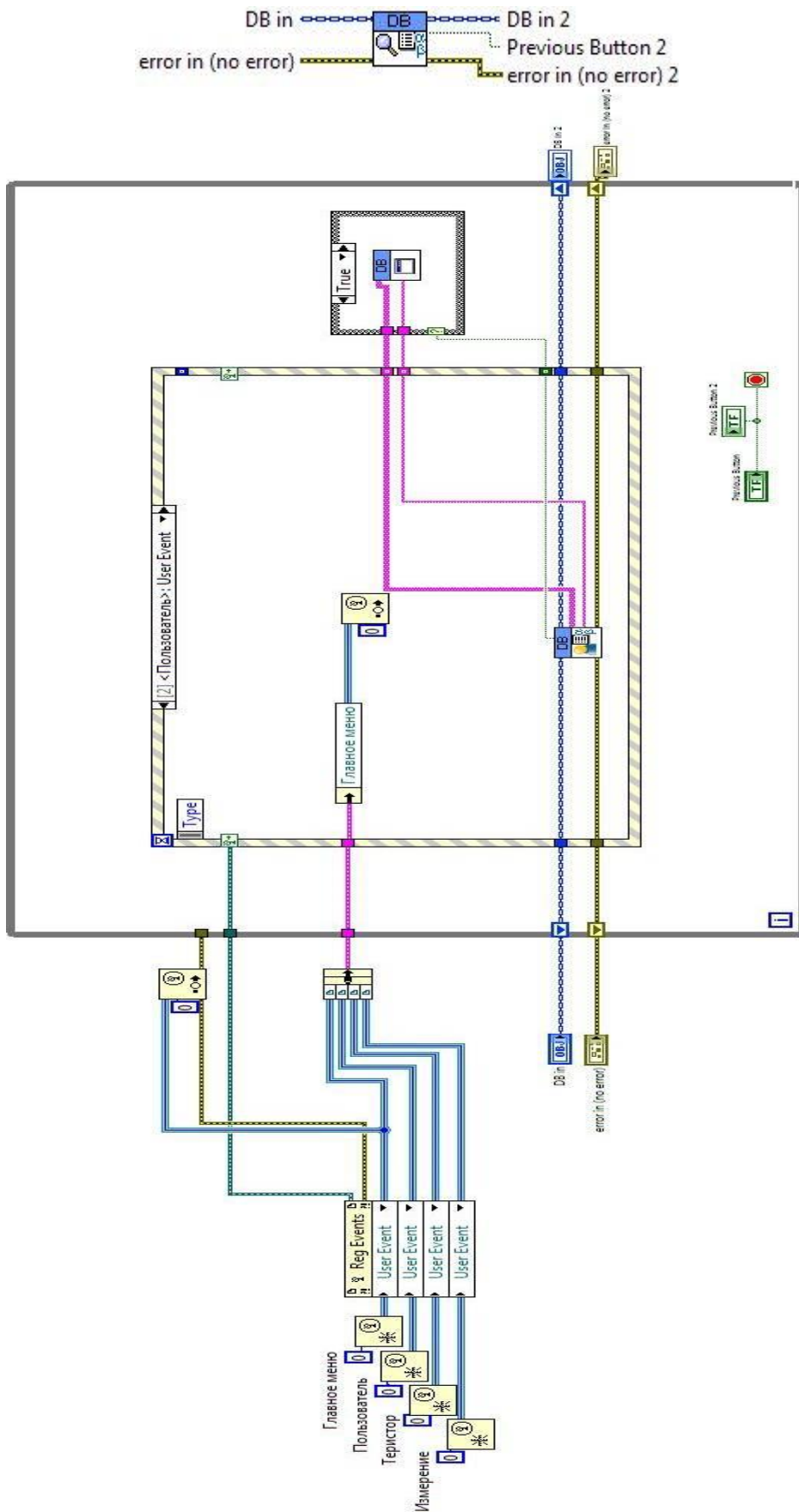


Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

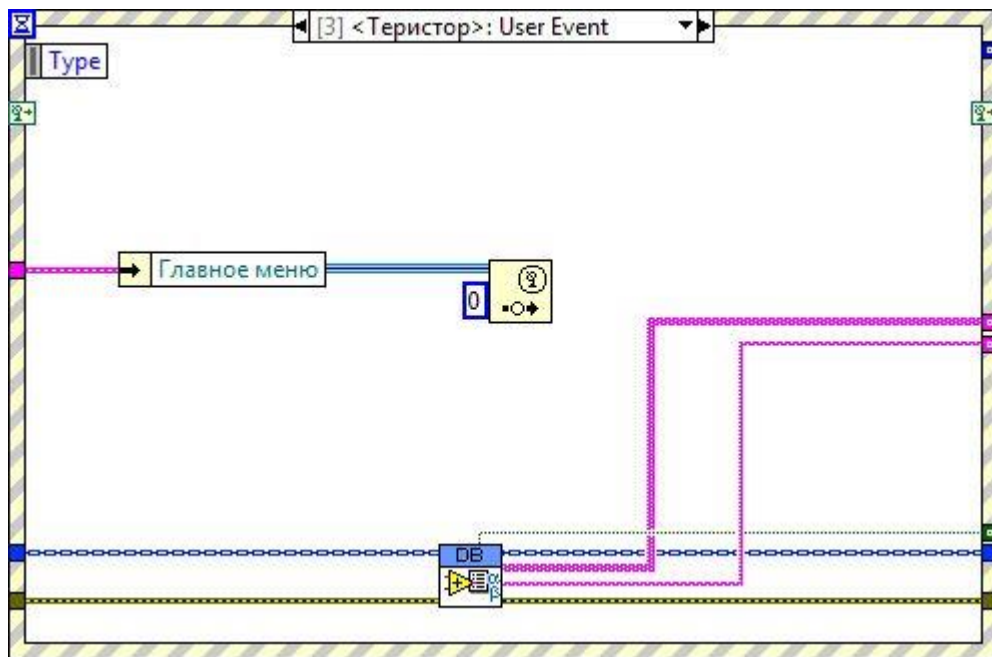
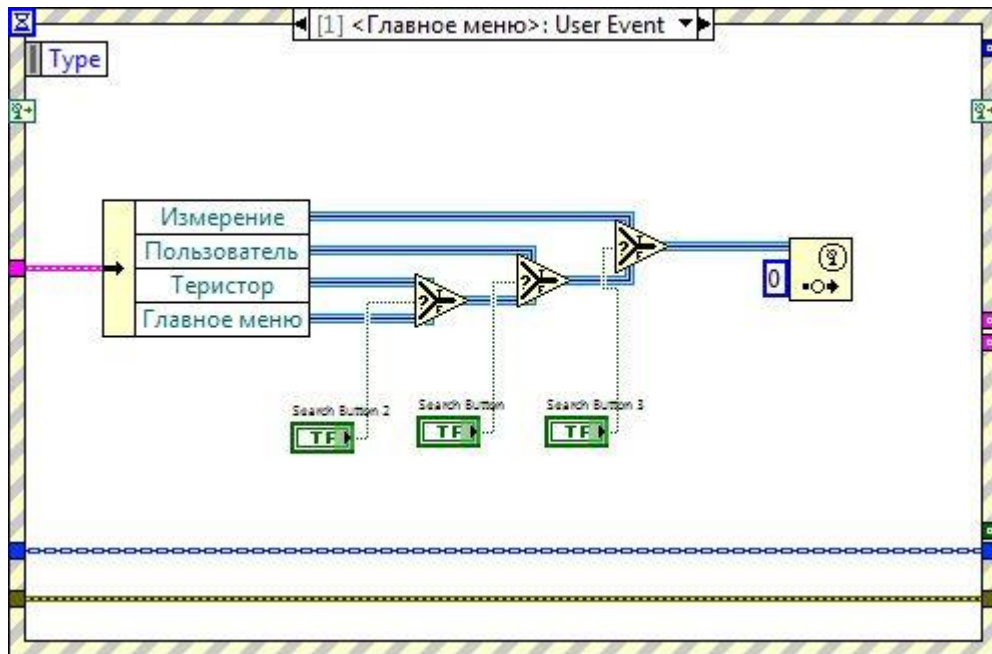
Лист

66



Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

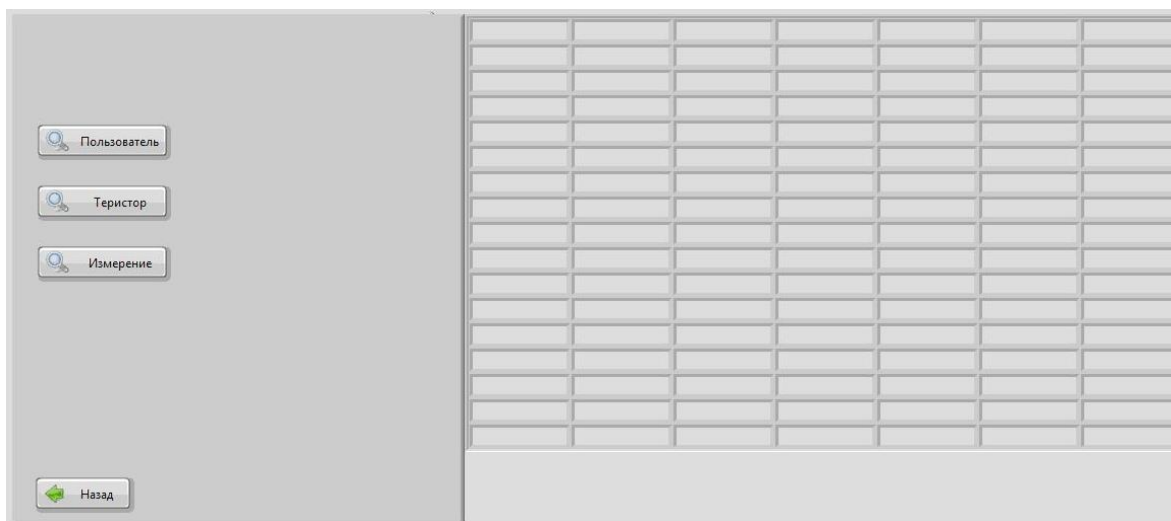
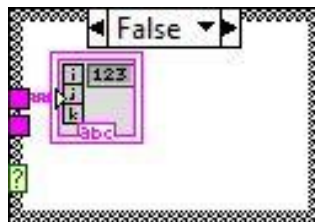
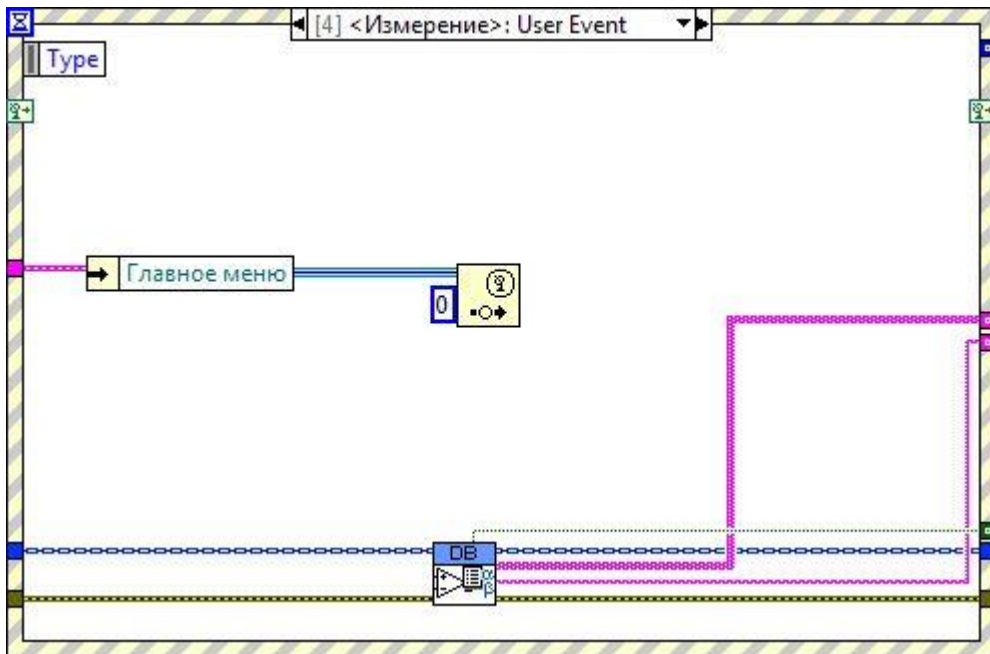
БР-02069964-12.03.01-10-18



Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18



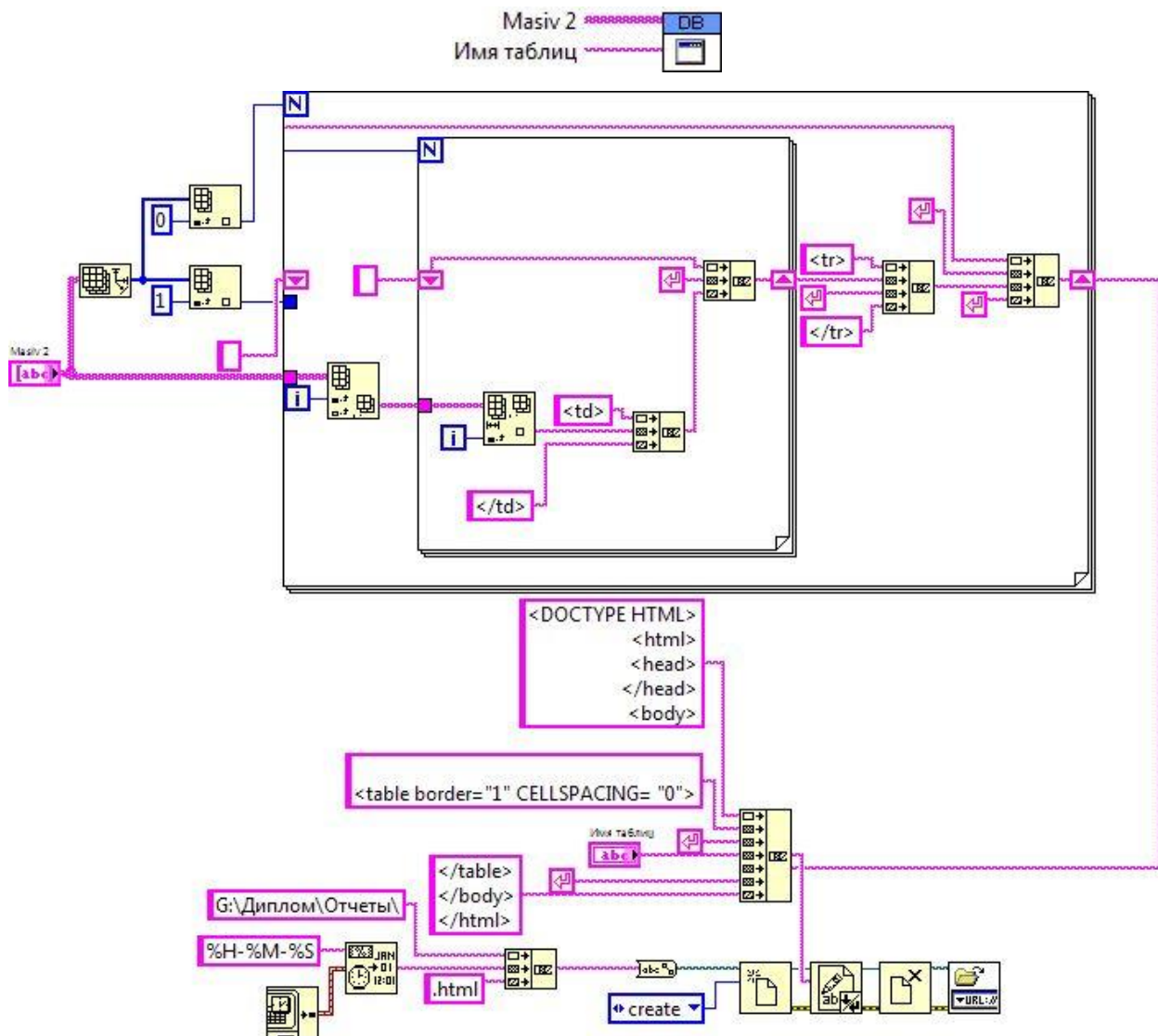


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

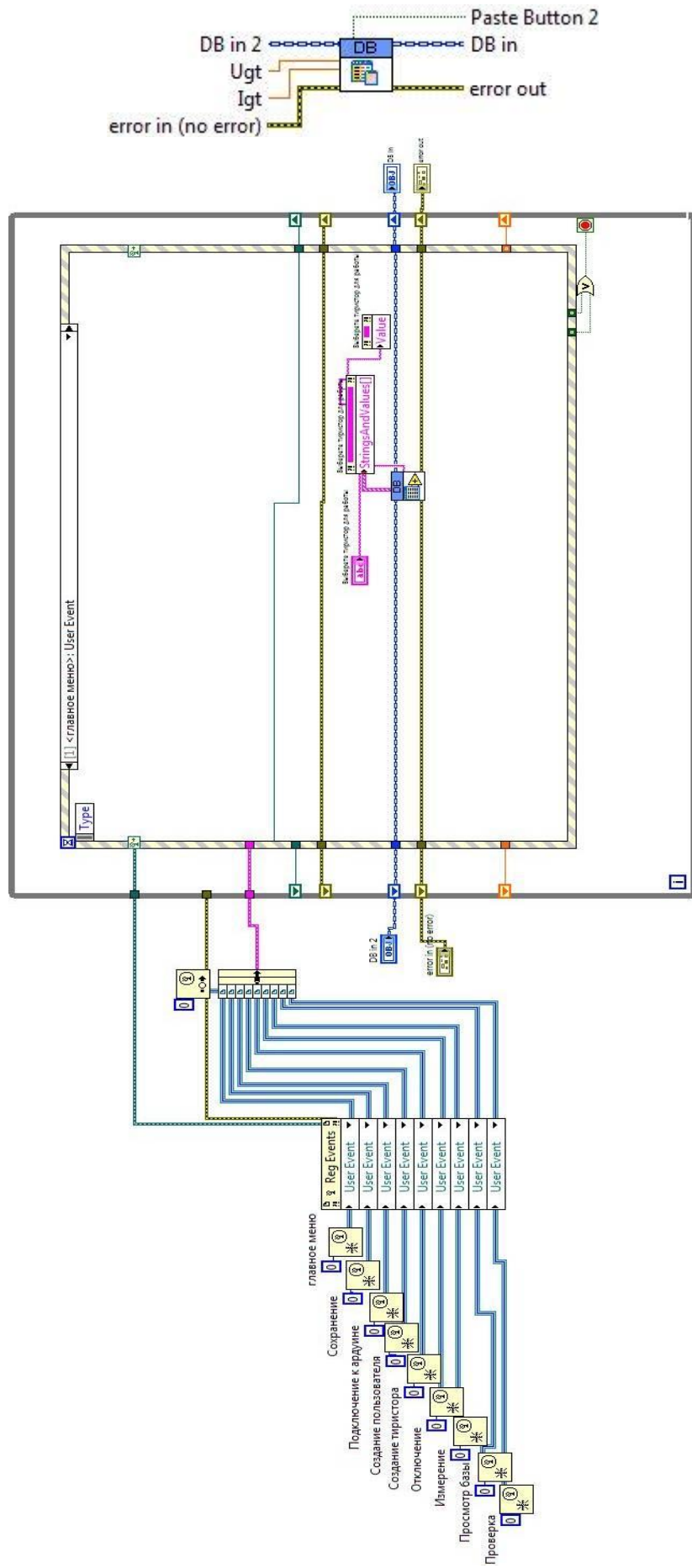
Лист

69



Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

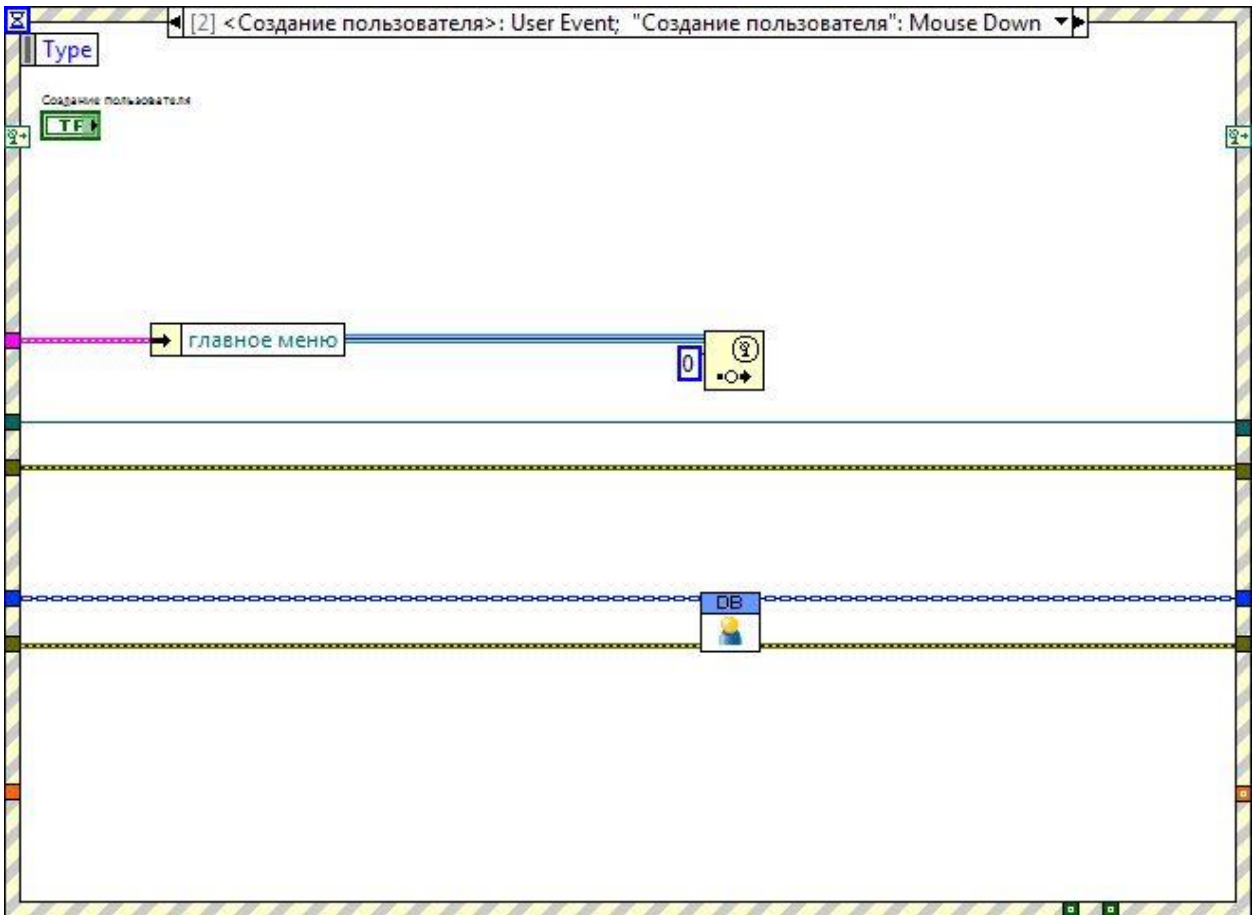
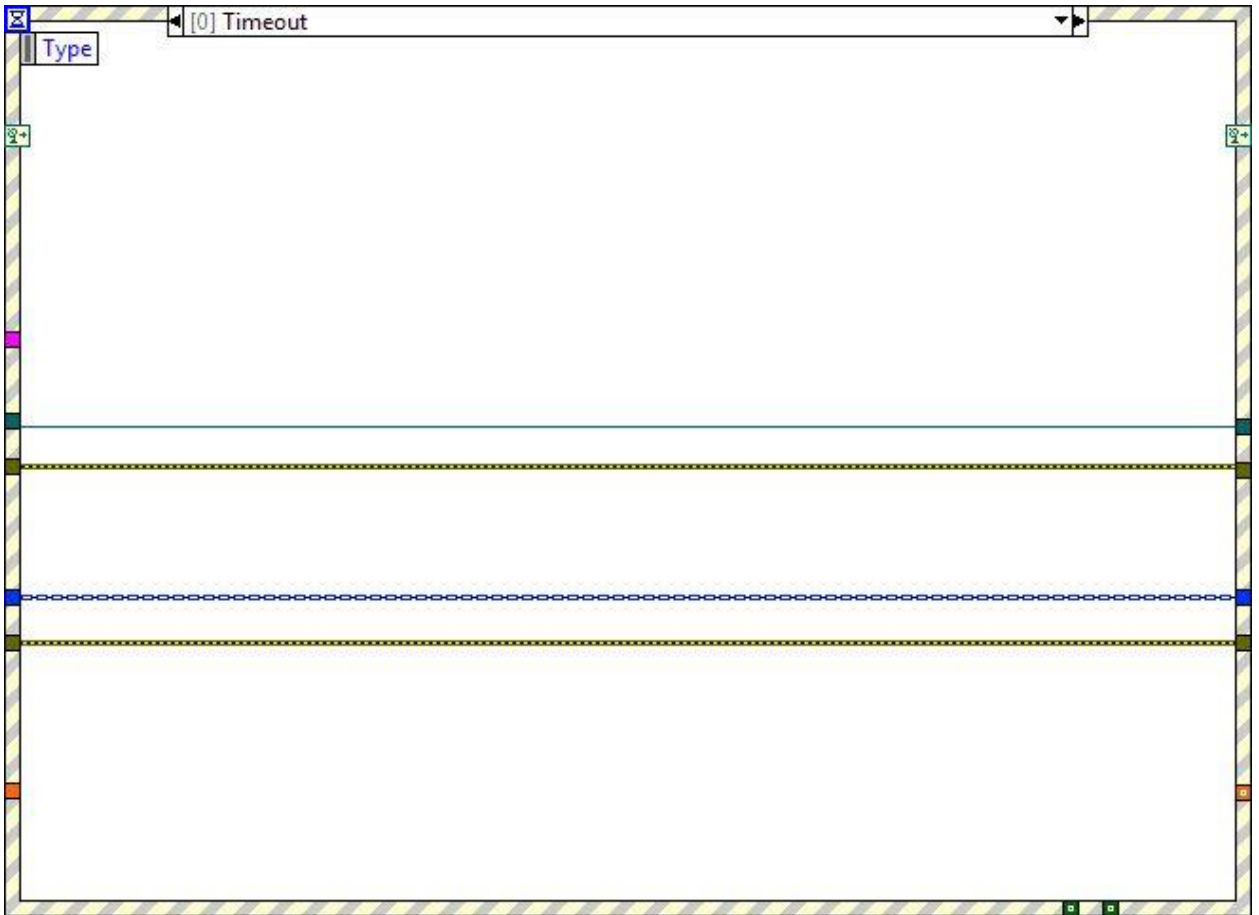


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

71

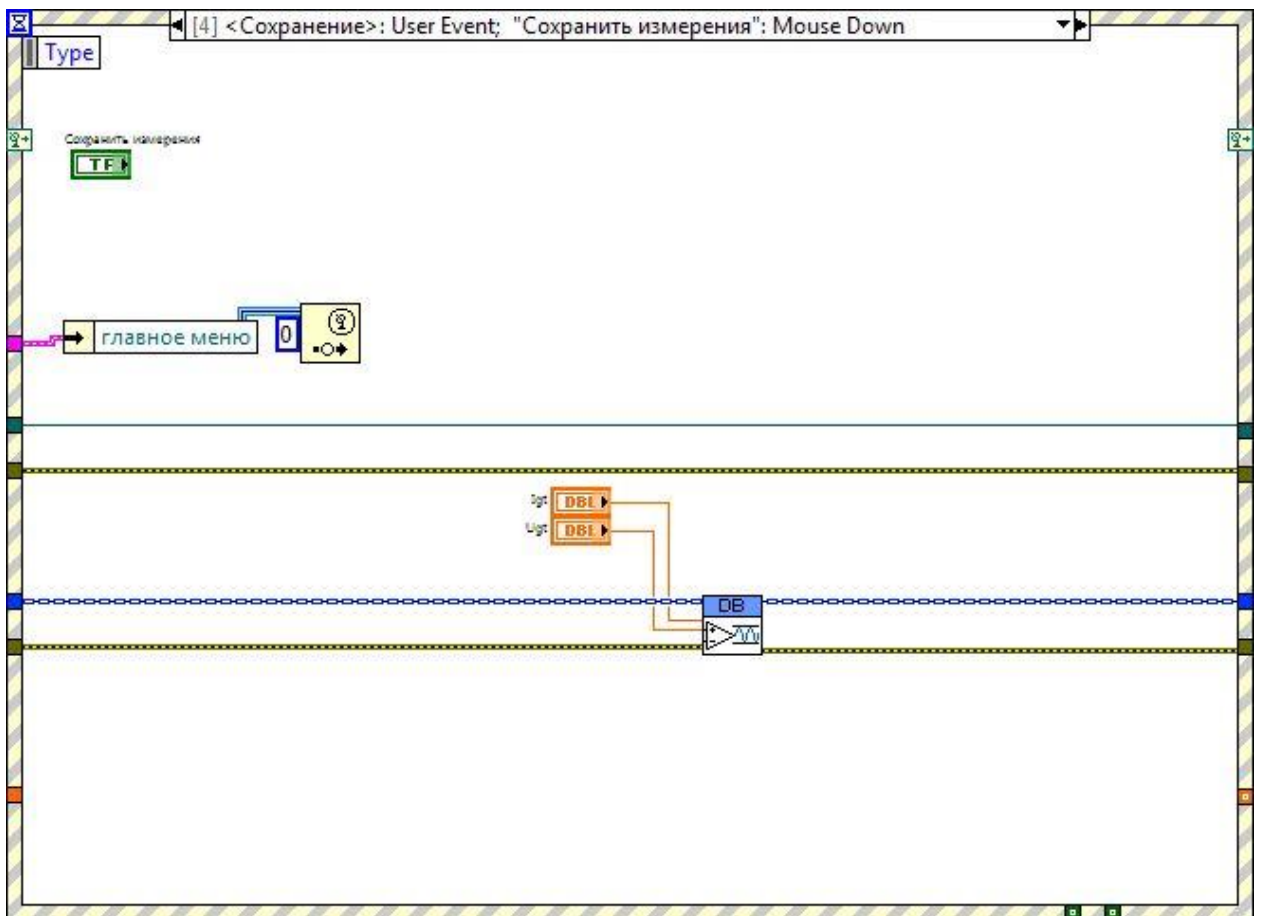
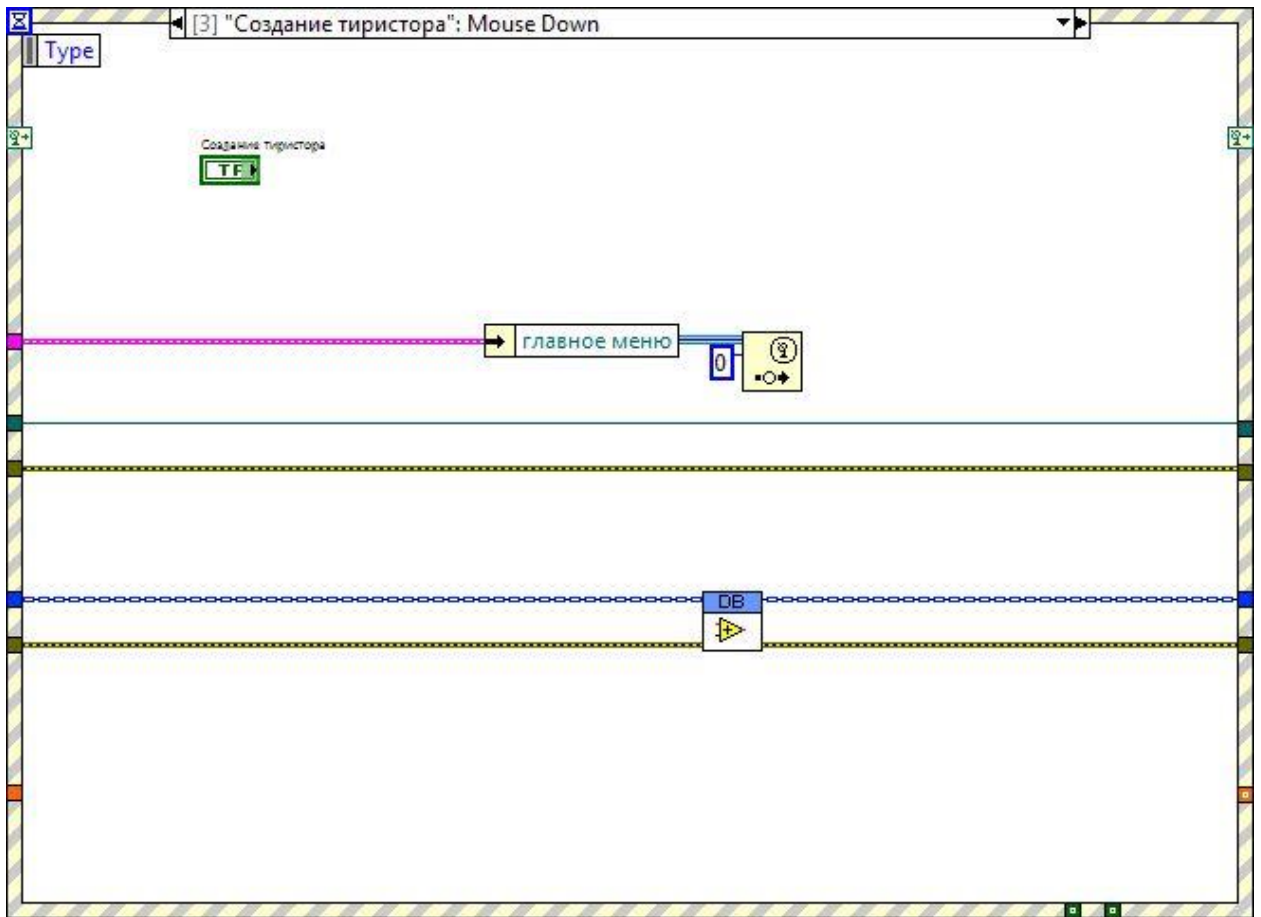


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

72

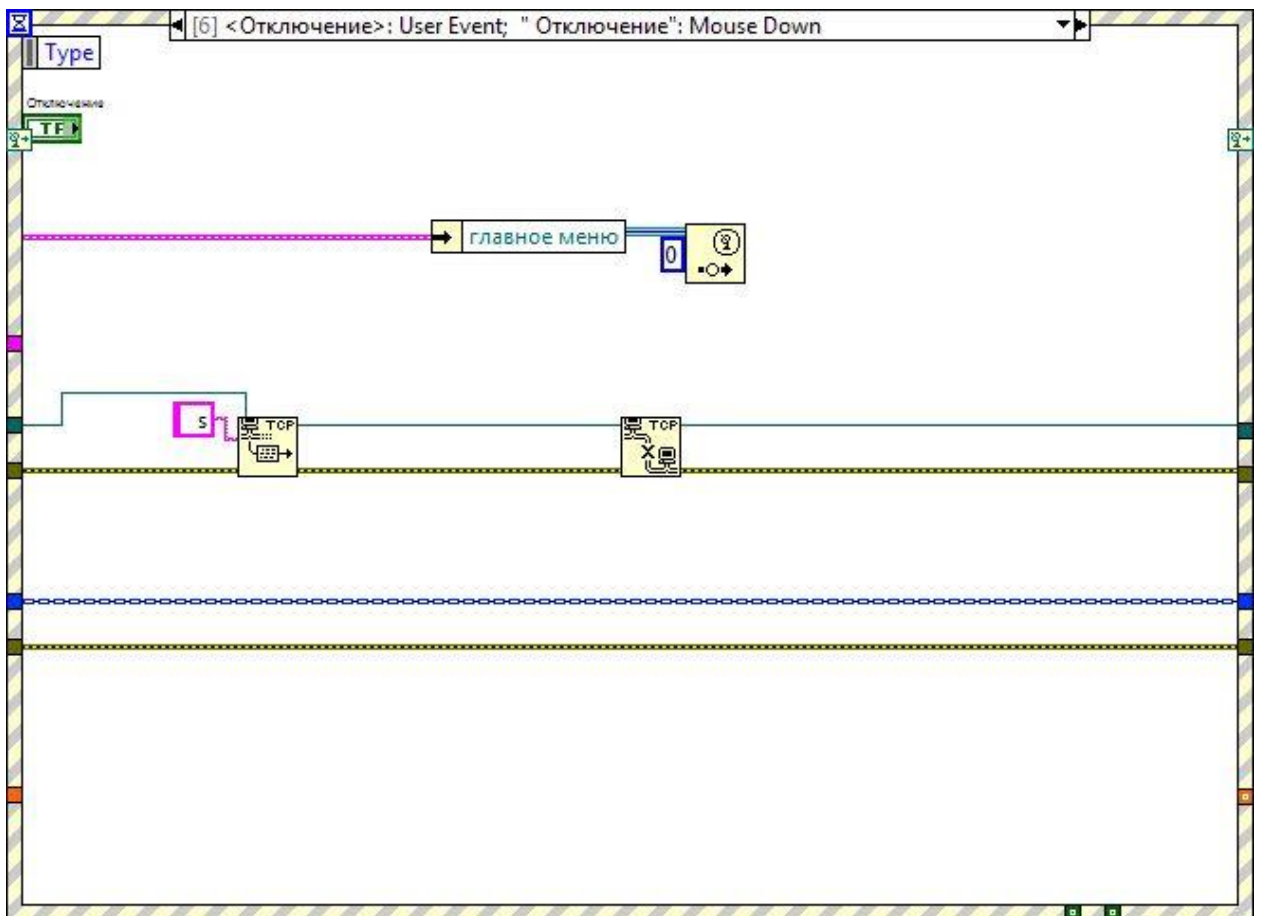
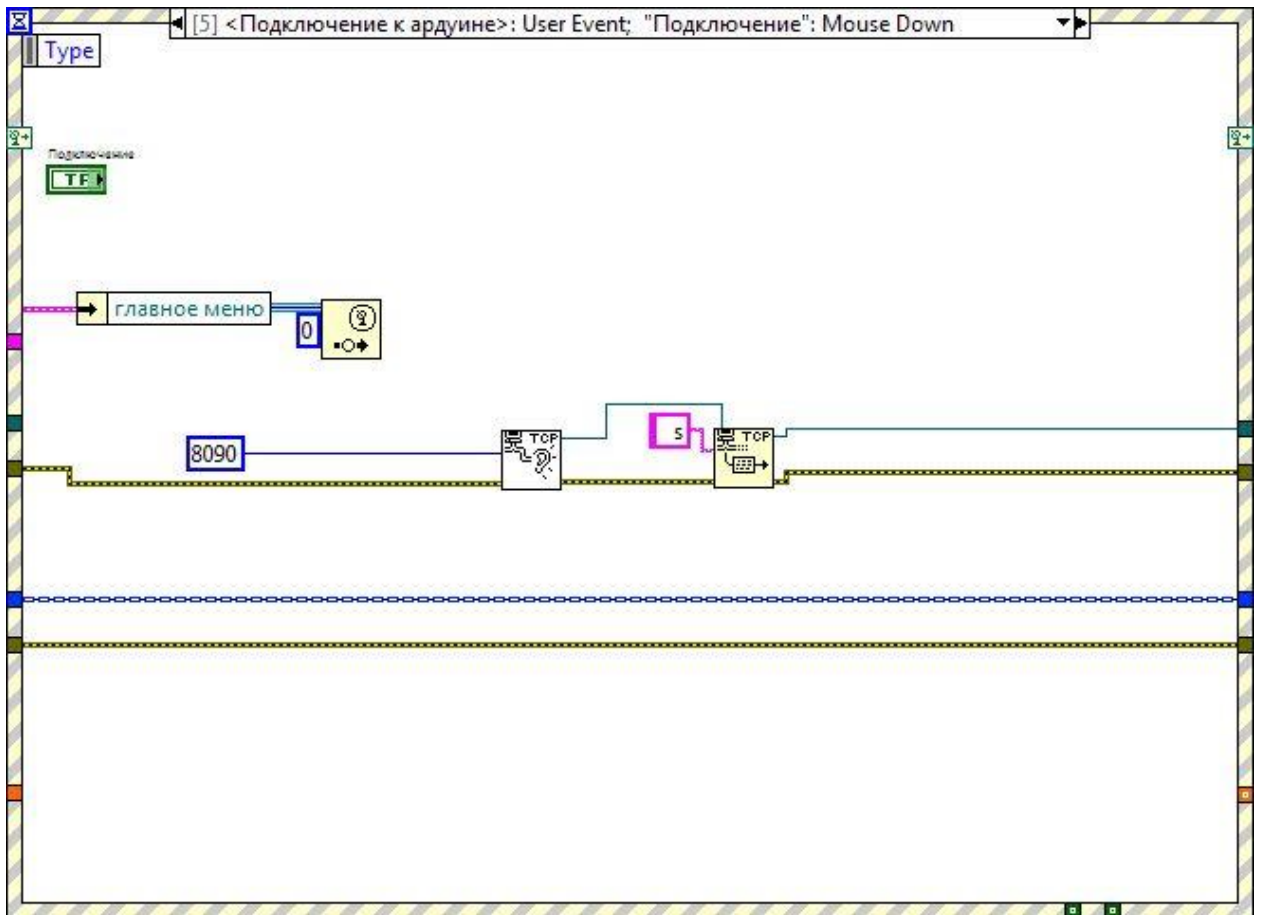


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

73

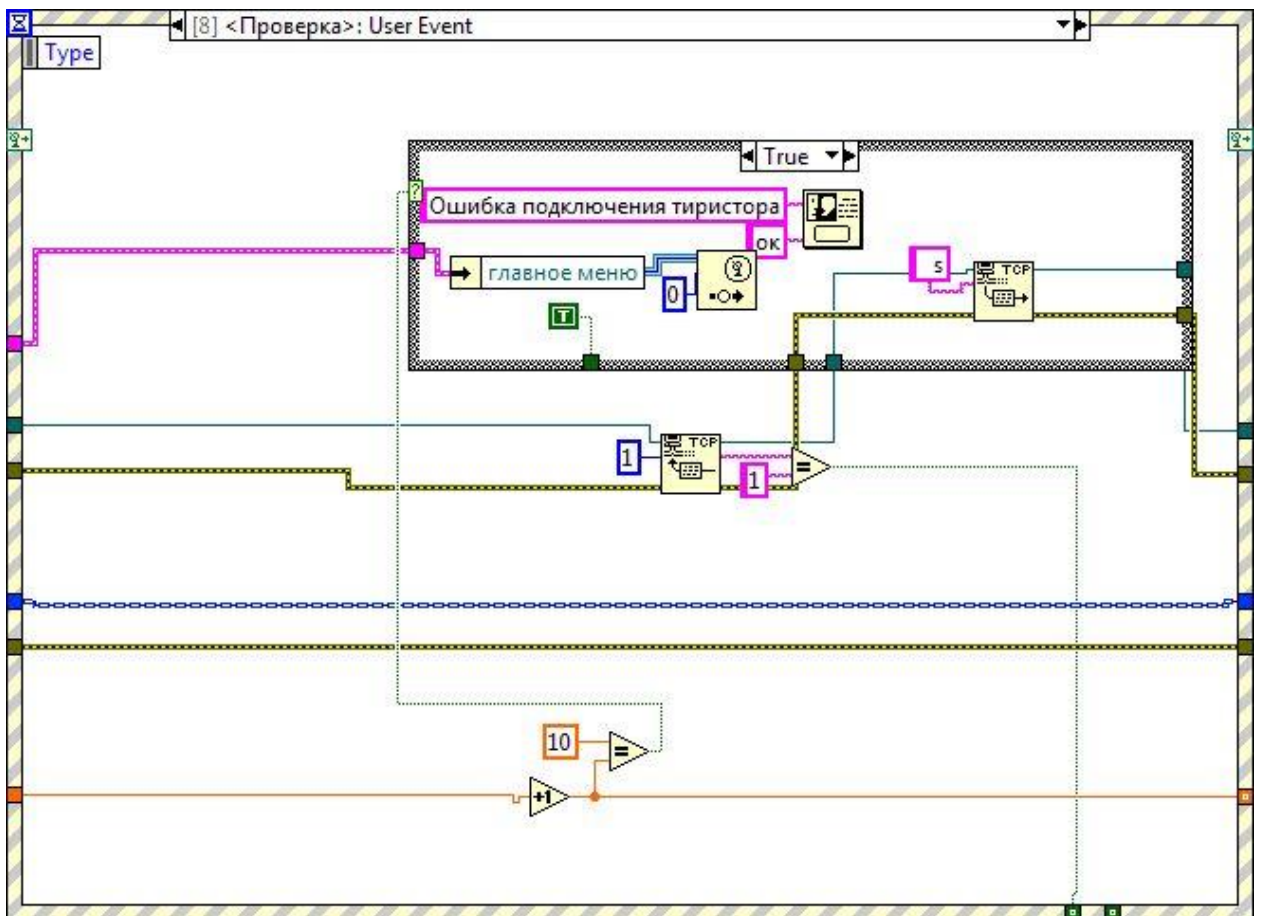
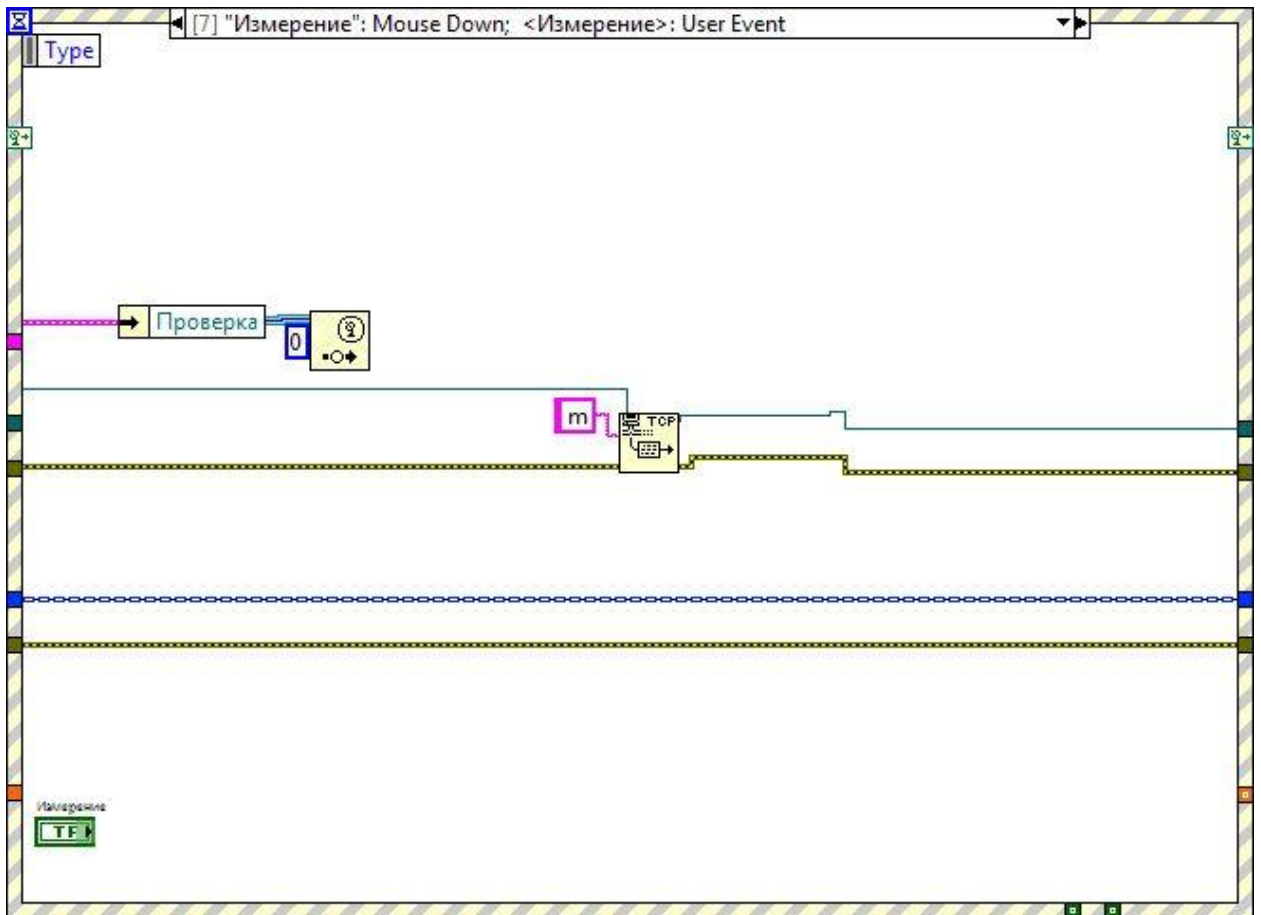


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

74

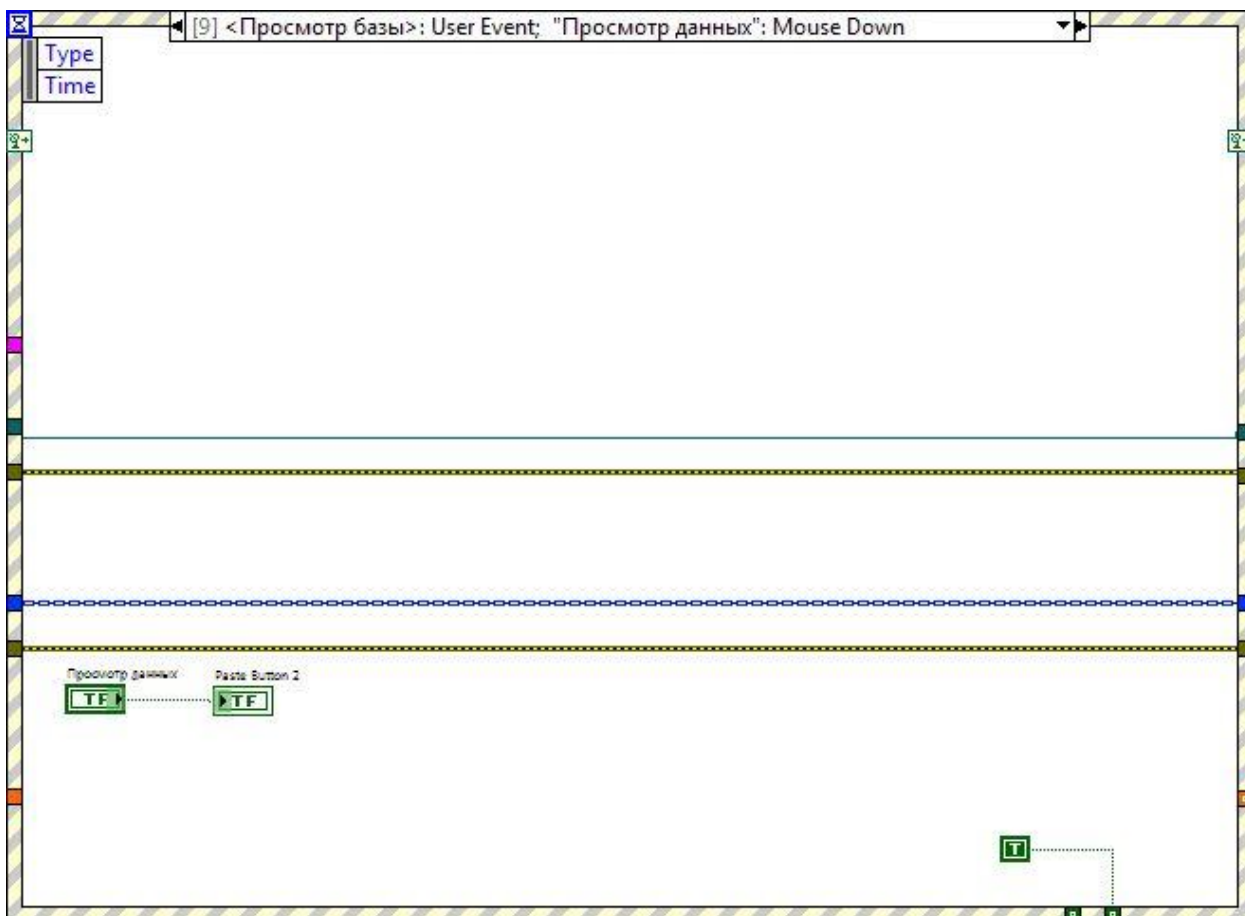
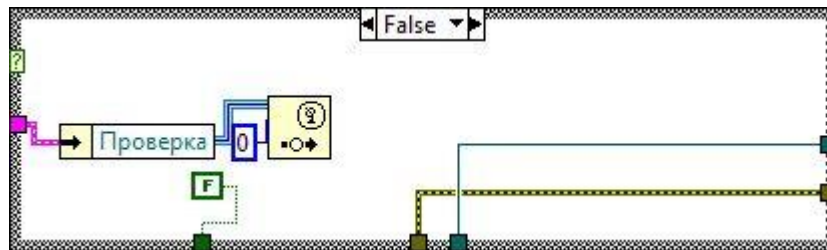


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

75



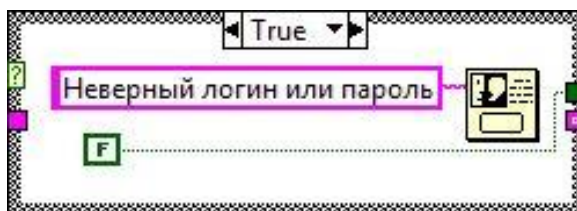
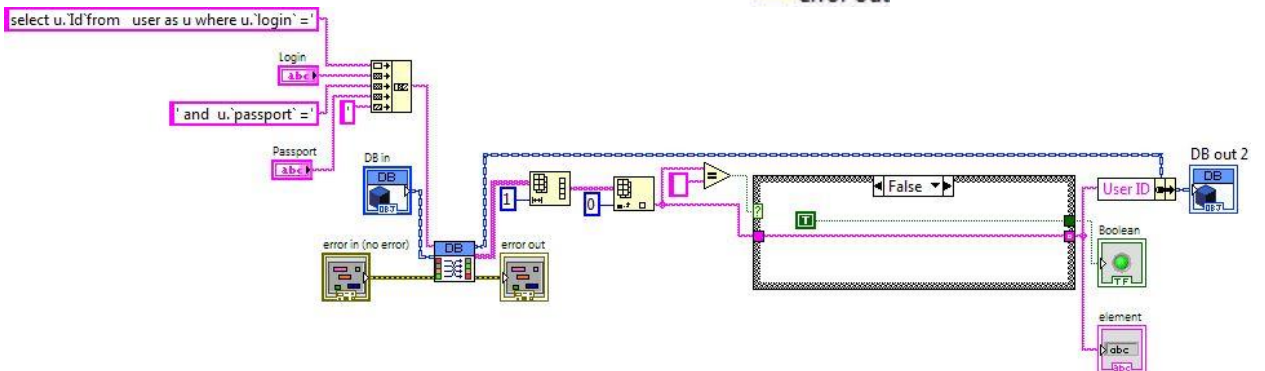
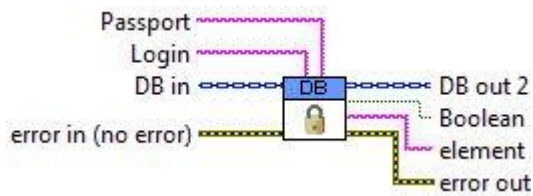
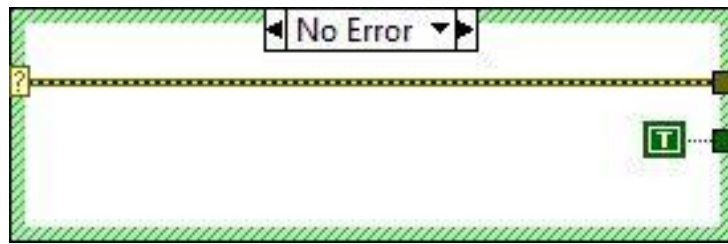
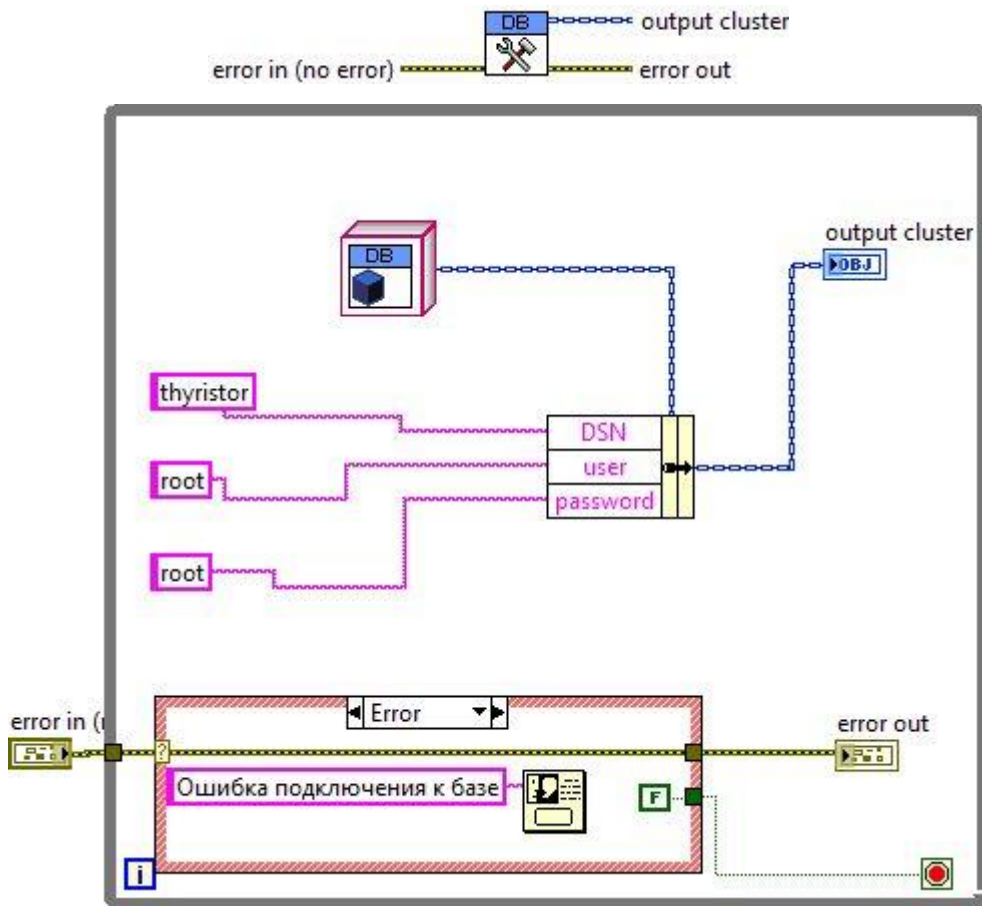
Control panel interface showing various buttons and input fields:

- Ugt: 0
- Igt: 0
- Подключение (On)
- Отключение (Off)
- Измерение (Measurement)
- Создание тиристора (Create thyristor)
- Создание пользователя (Create user)
- Выберете тиристор для работы (Select thyristor for work)
- Просмотр базы (View database)
- Сохранить измерение (Save measurement)

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

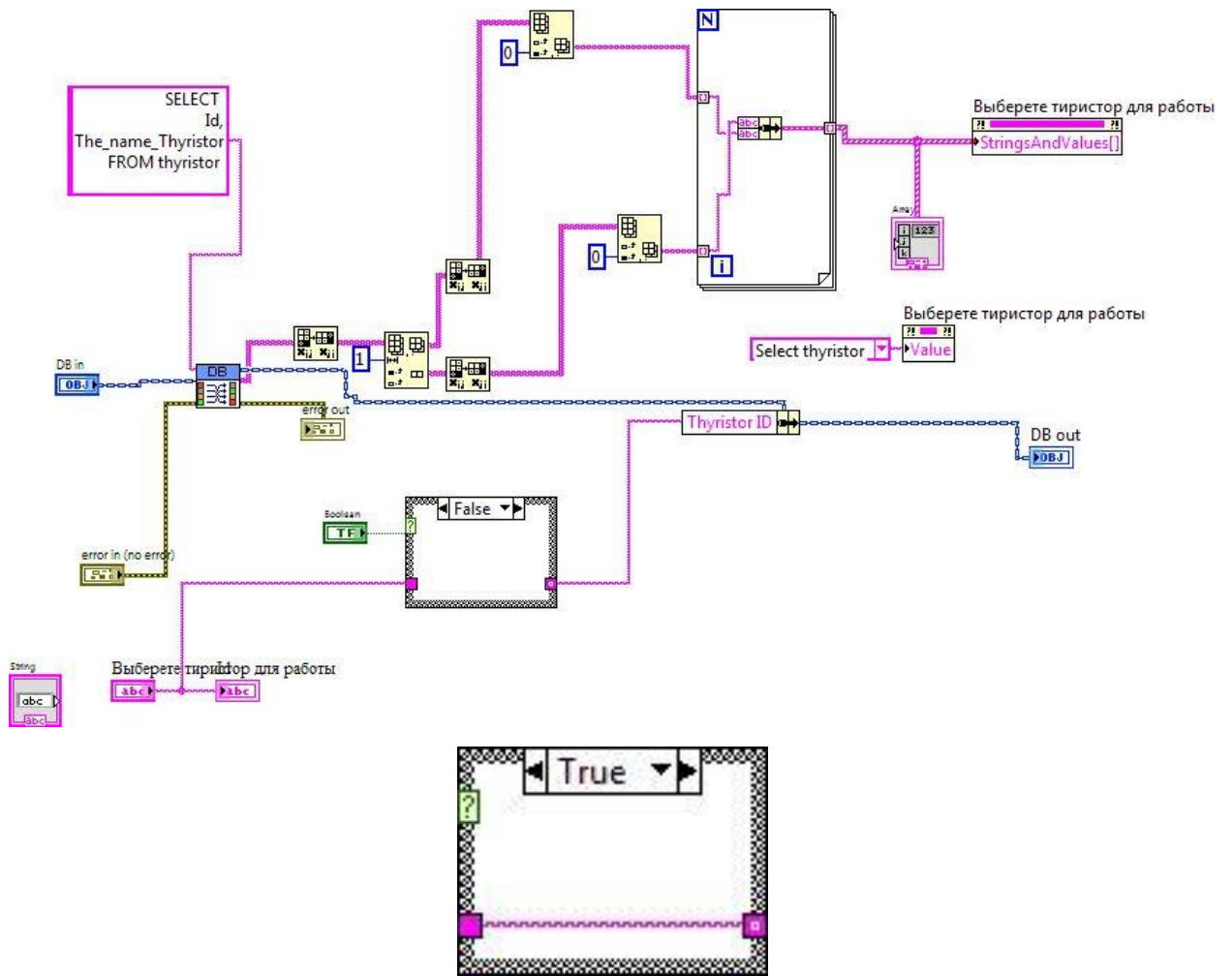




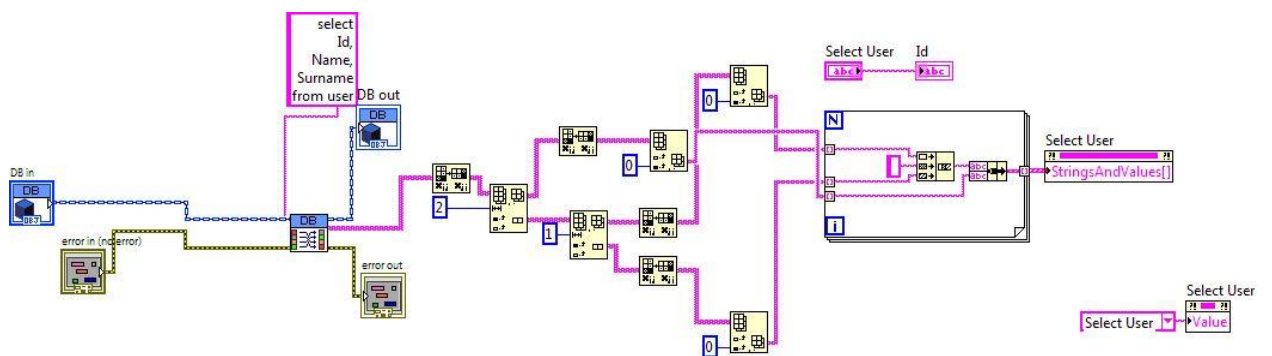
Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Выберете тиристор для работы  
 Array  
 DB in DB out  
 Boolean Id  
 error in (no error) error out



Select User  
 DB in DB out  
 error in (no error) Id error out

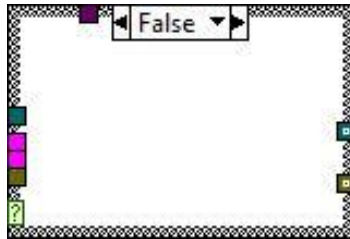
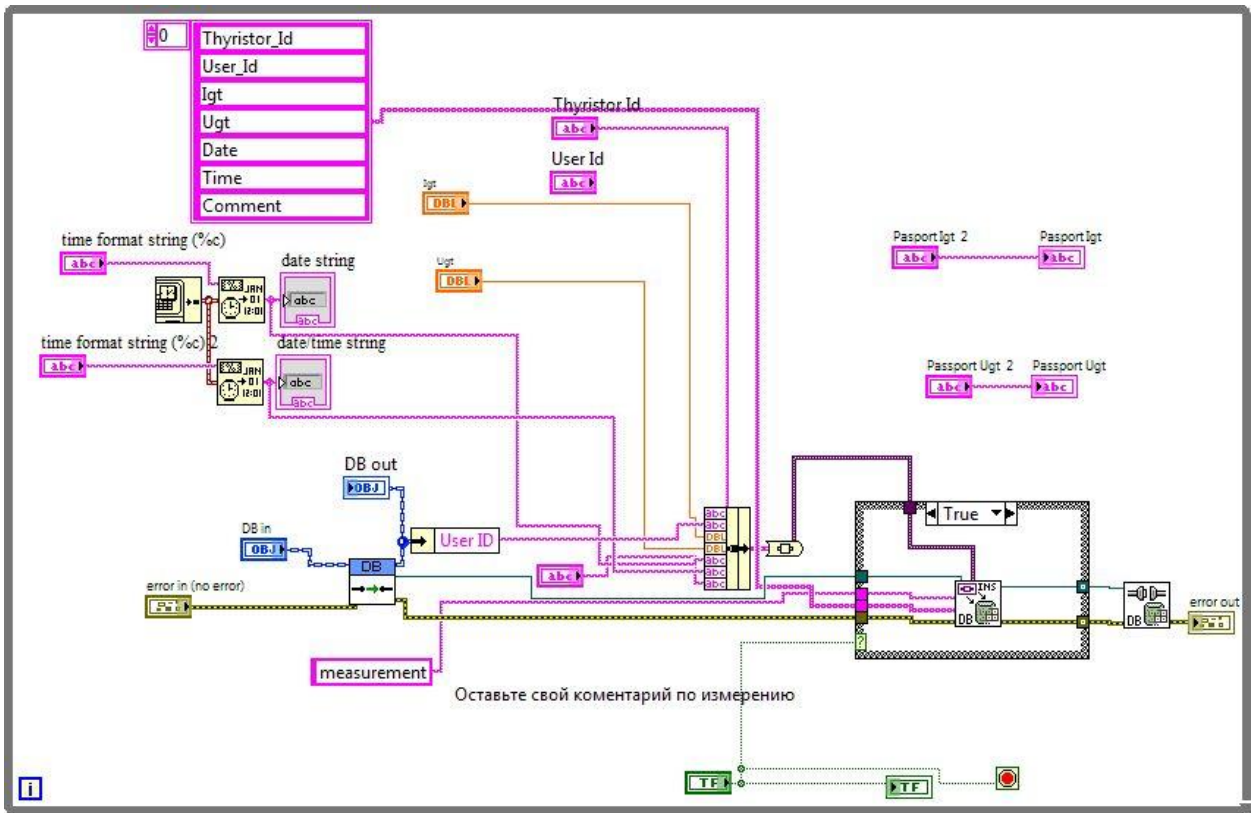
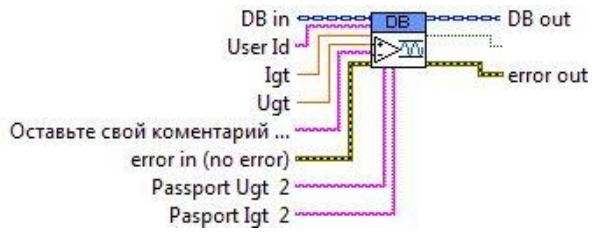


Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

Лист

78



Igt  Pasport Igt

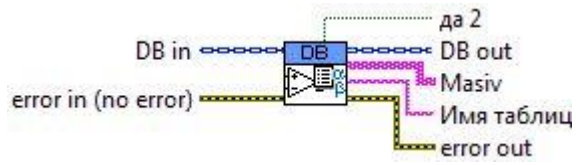
Ugt  Pasport Ugt

Оставьте свой комментарий по измерению

Save

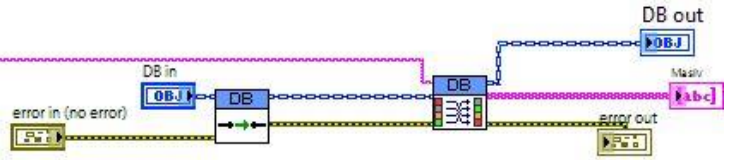
Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

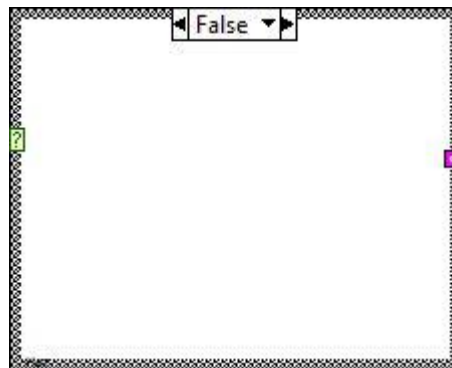
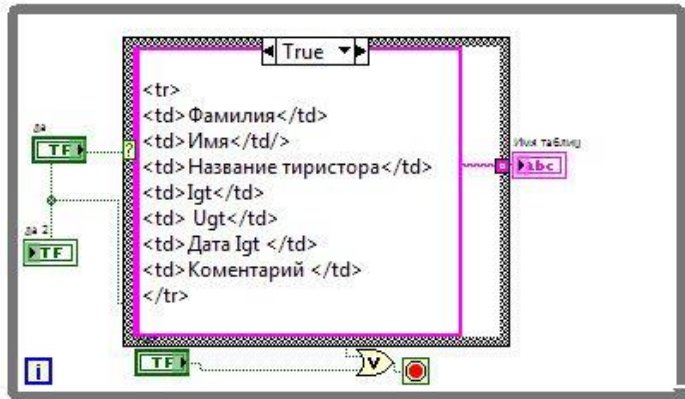


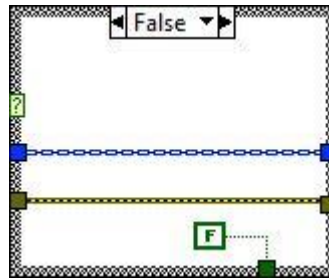
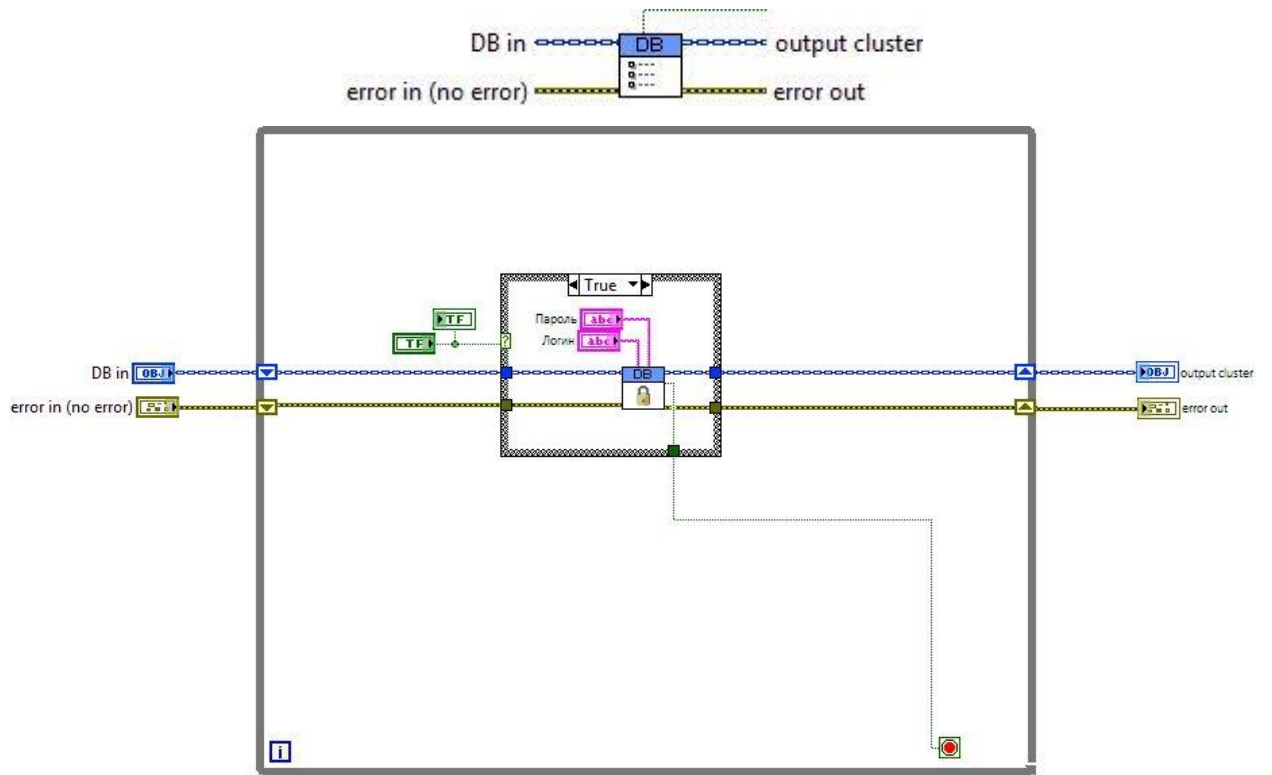
```

select
u.Surname,
u.Name,
t.The_name_Thyristor,
m.Igt,
m.ugt,
m. Date,
m. Comment
from measurement as m join user as u join
thyristor as t where
m.User_Id = u.id AND
  
```



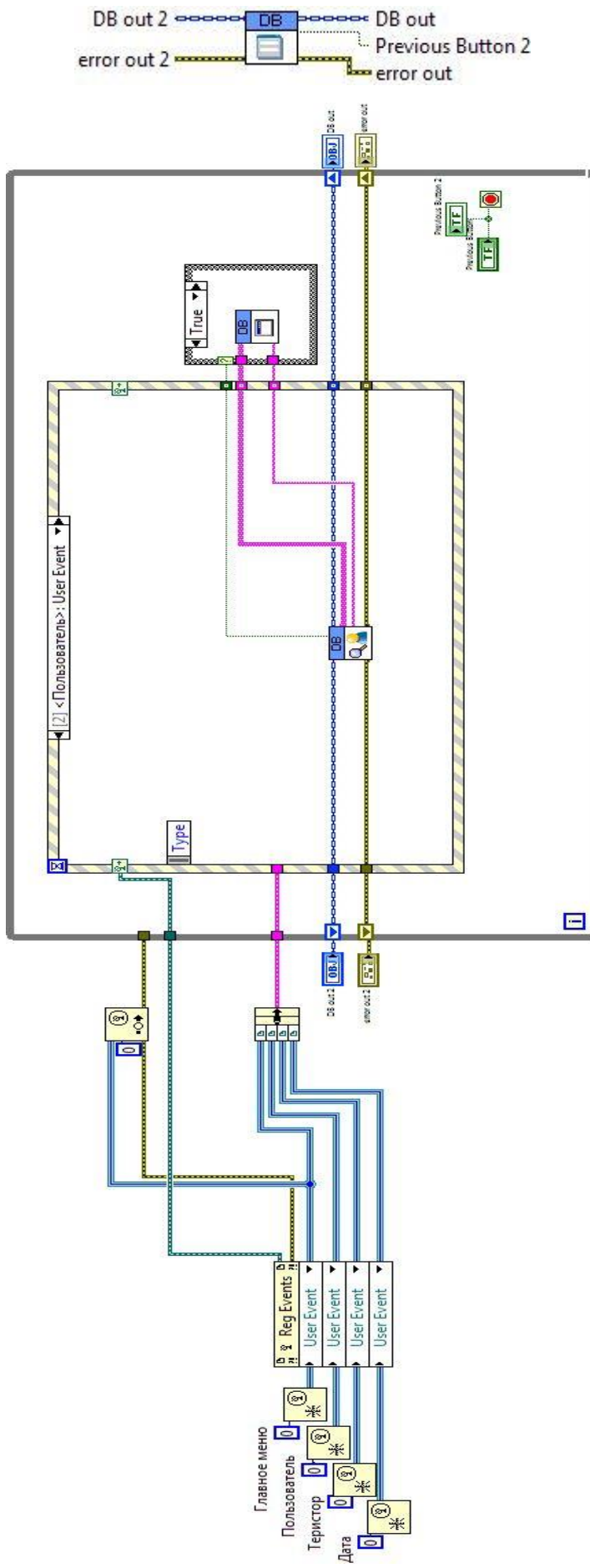
Формировать отчет ?





Логин

Пароль

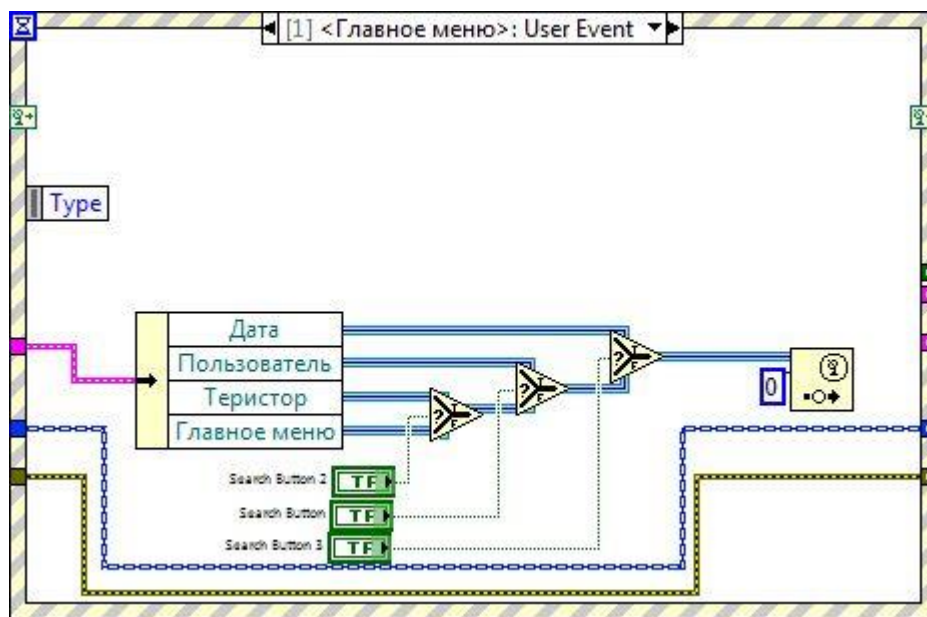
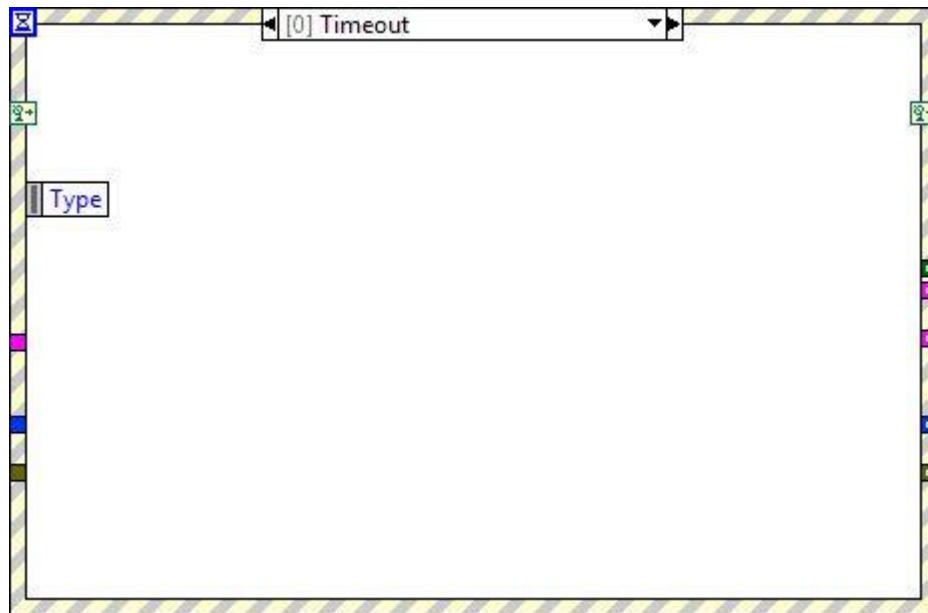


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

82

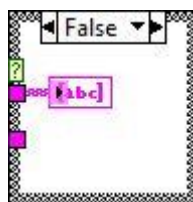
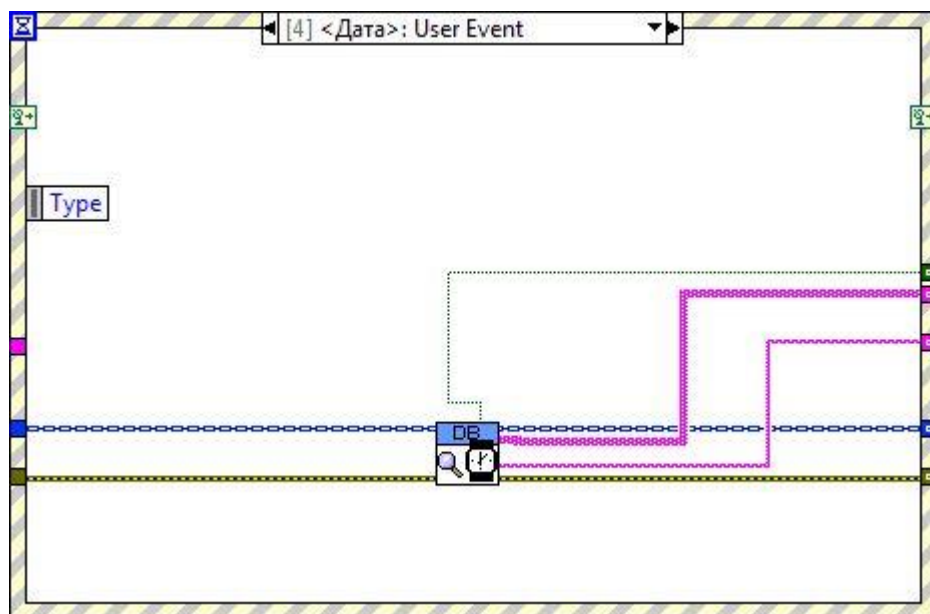
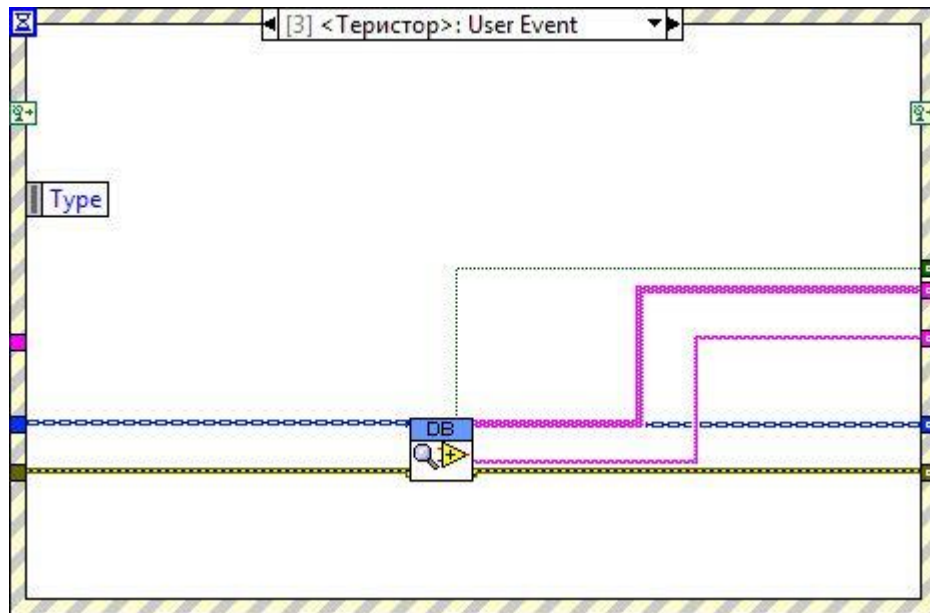


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

83



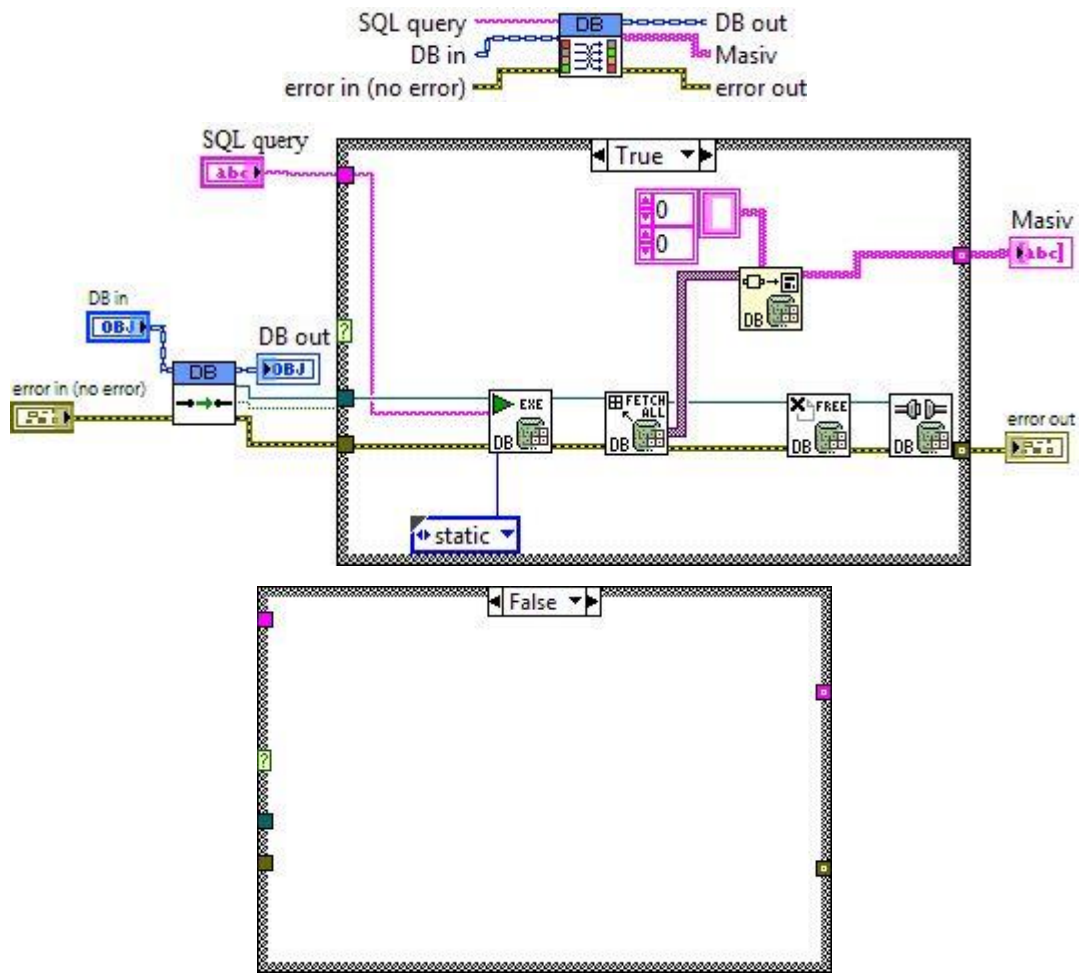
Поиск Пользователь

Поиск Теристор

Поиск Дата

Назад



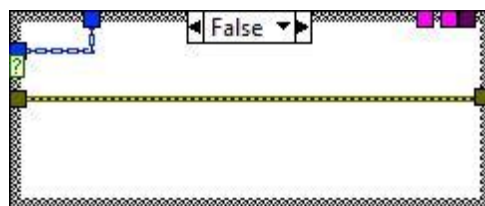
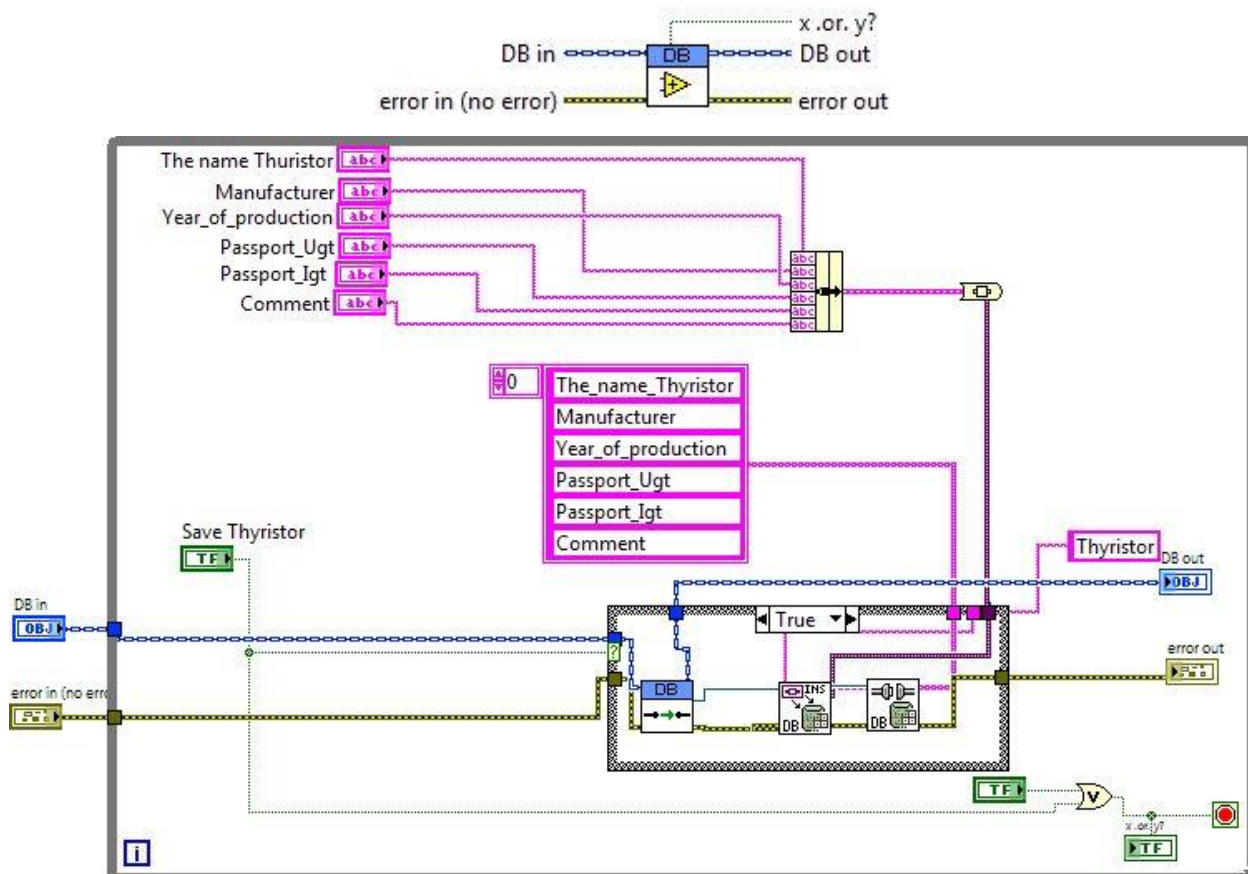



Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

Лист

85



Название теристора

Производитель

Год производства

Паспортное Ugt

Паспортное Igt

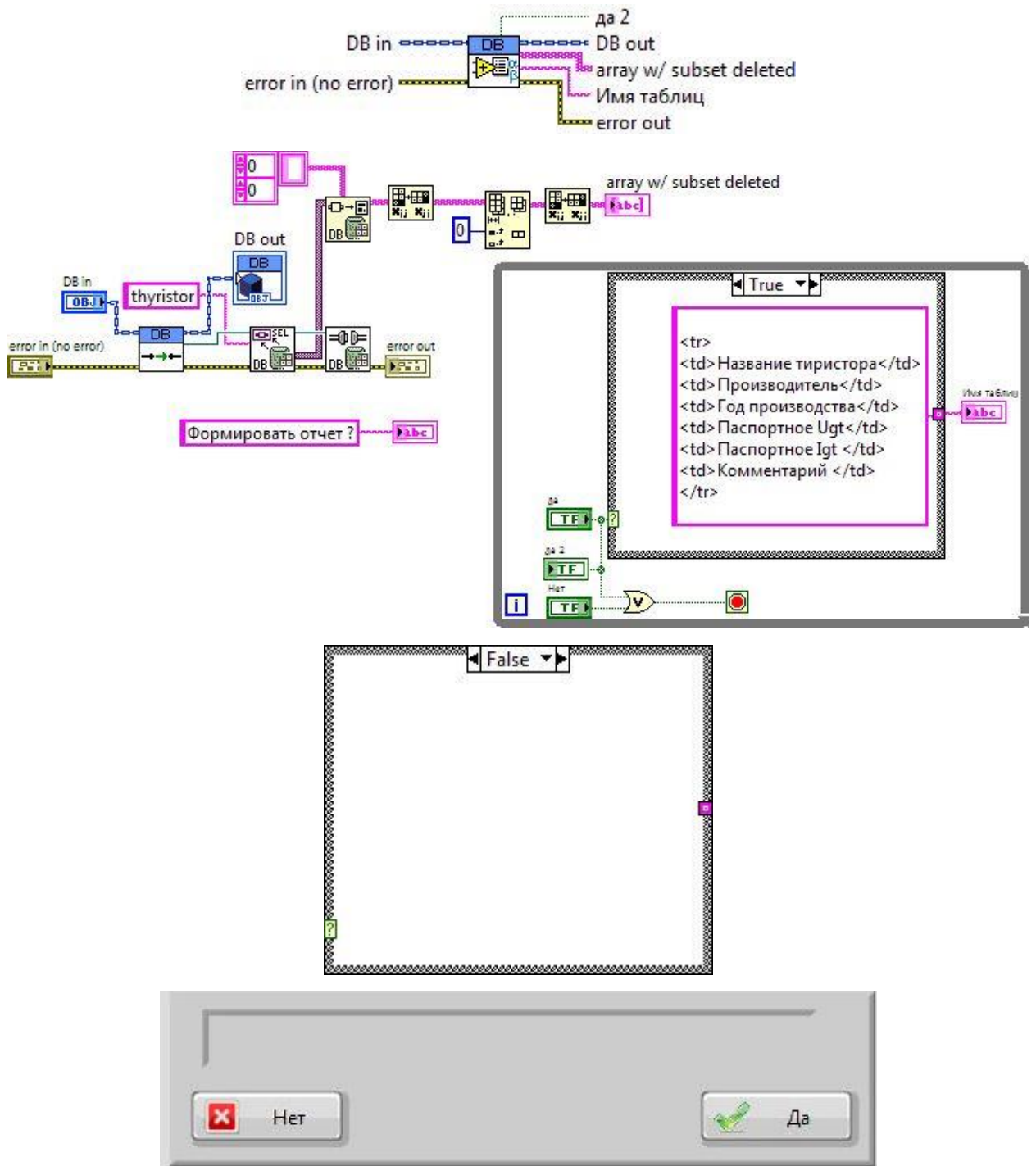
Комментарий

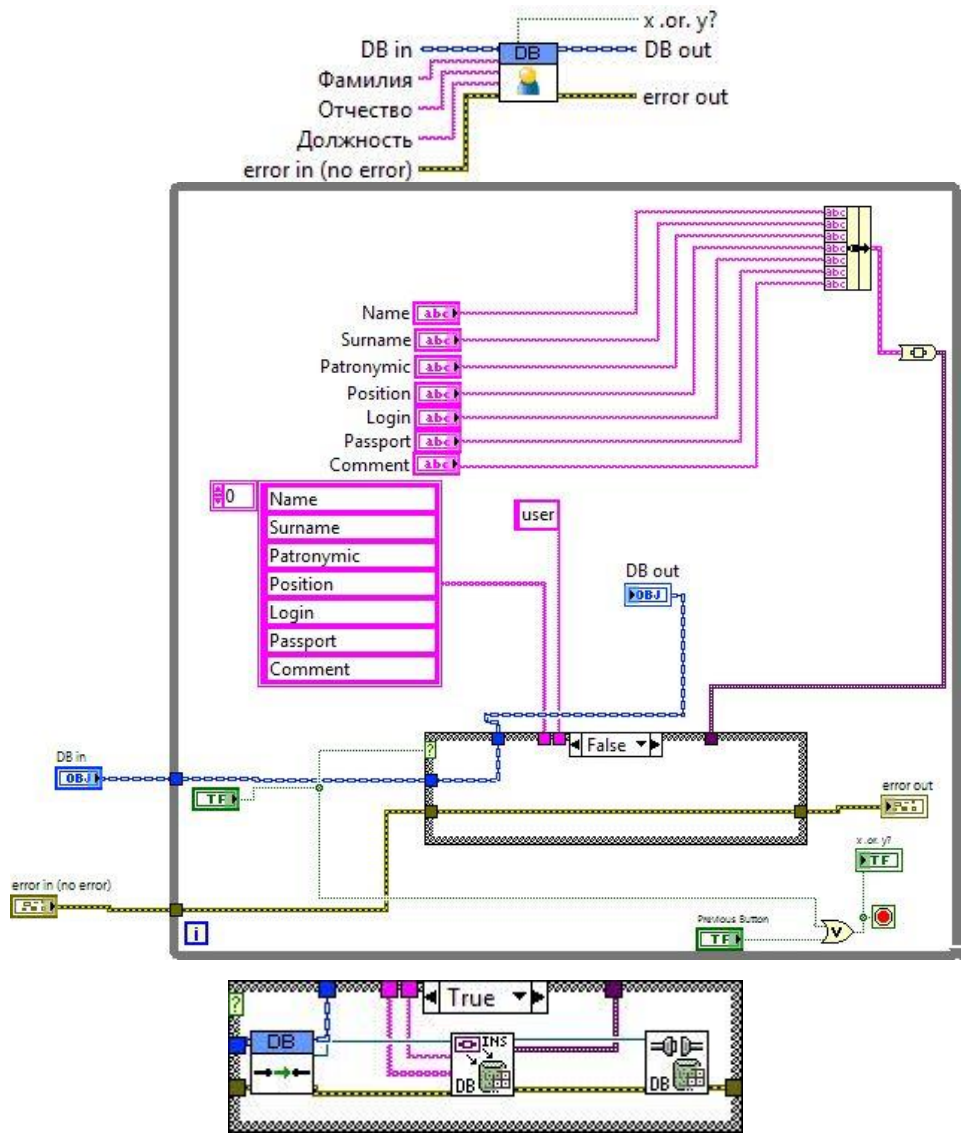
Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

Лист

86





Имя

Фамилия

Отчество

Логин

Пароль

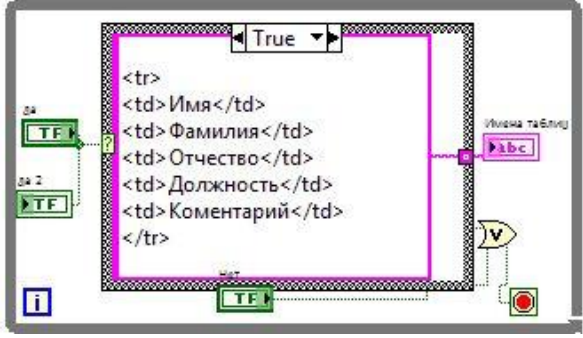
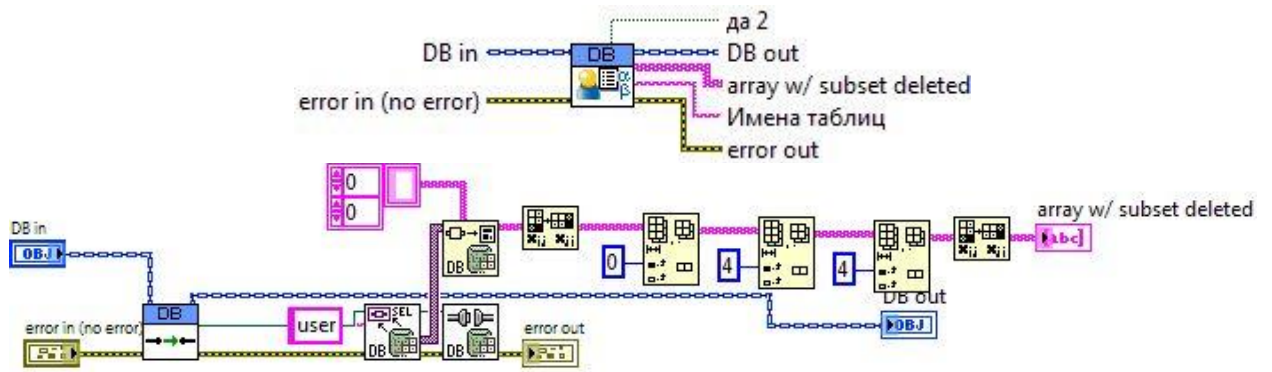
Должность

Оставьте свой комментарий (не обязательно)

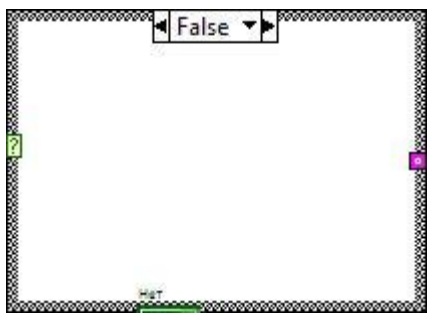
Назад Сохранить

Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

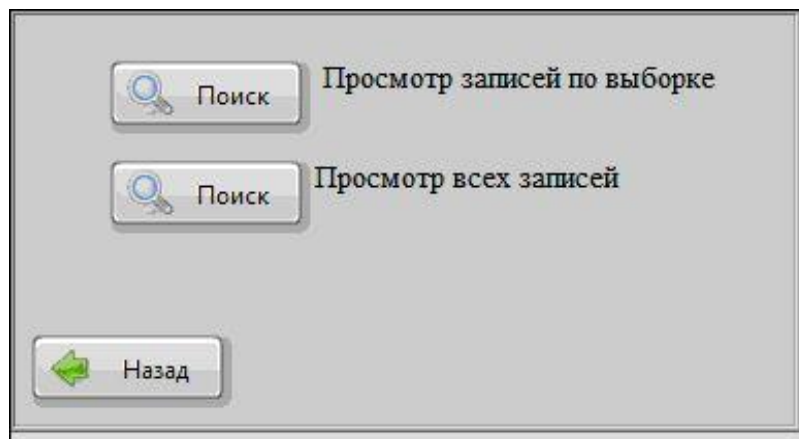
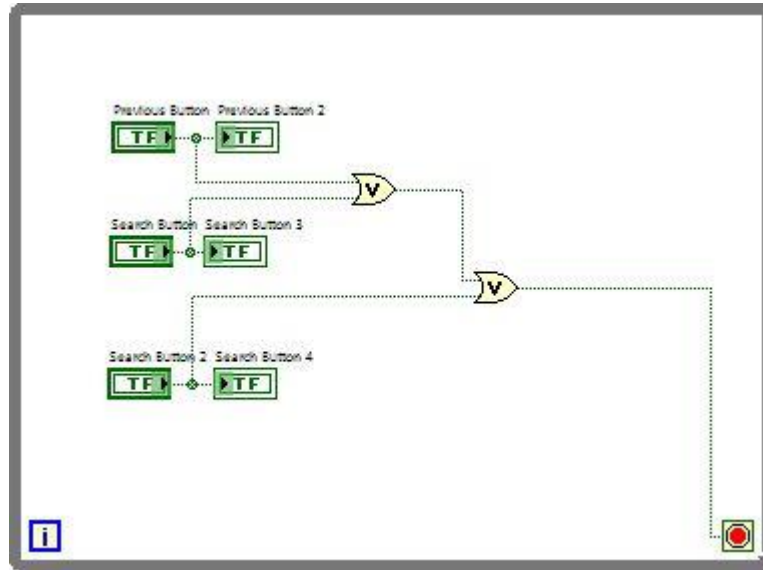
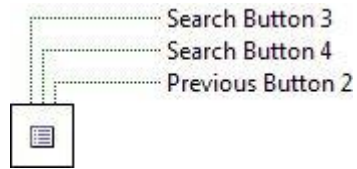


Формировать отчет? abc



✖ Нет

✔ Да

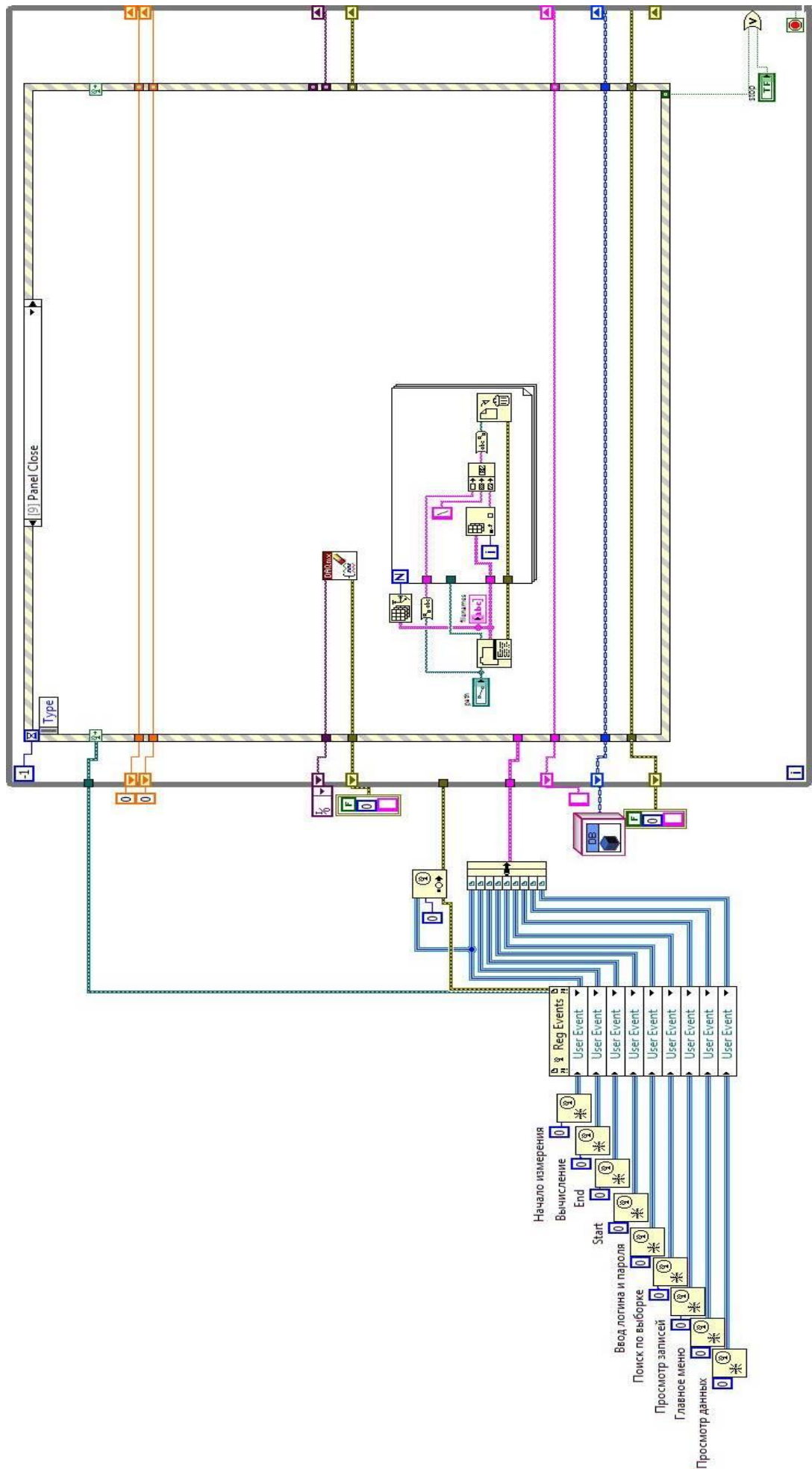


Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

БР-02069964-12.03.01-10-18

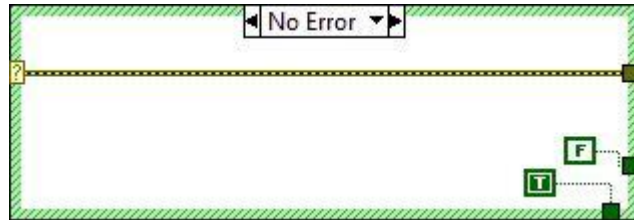
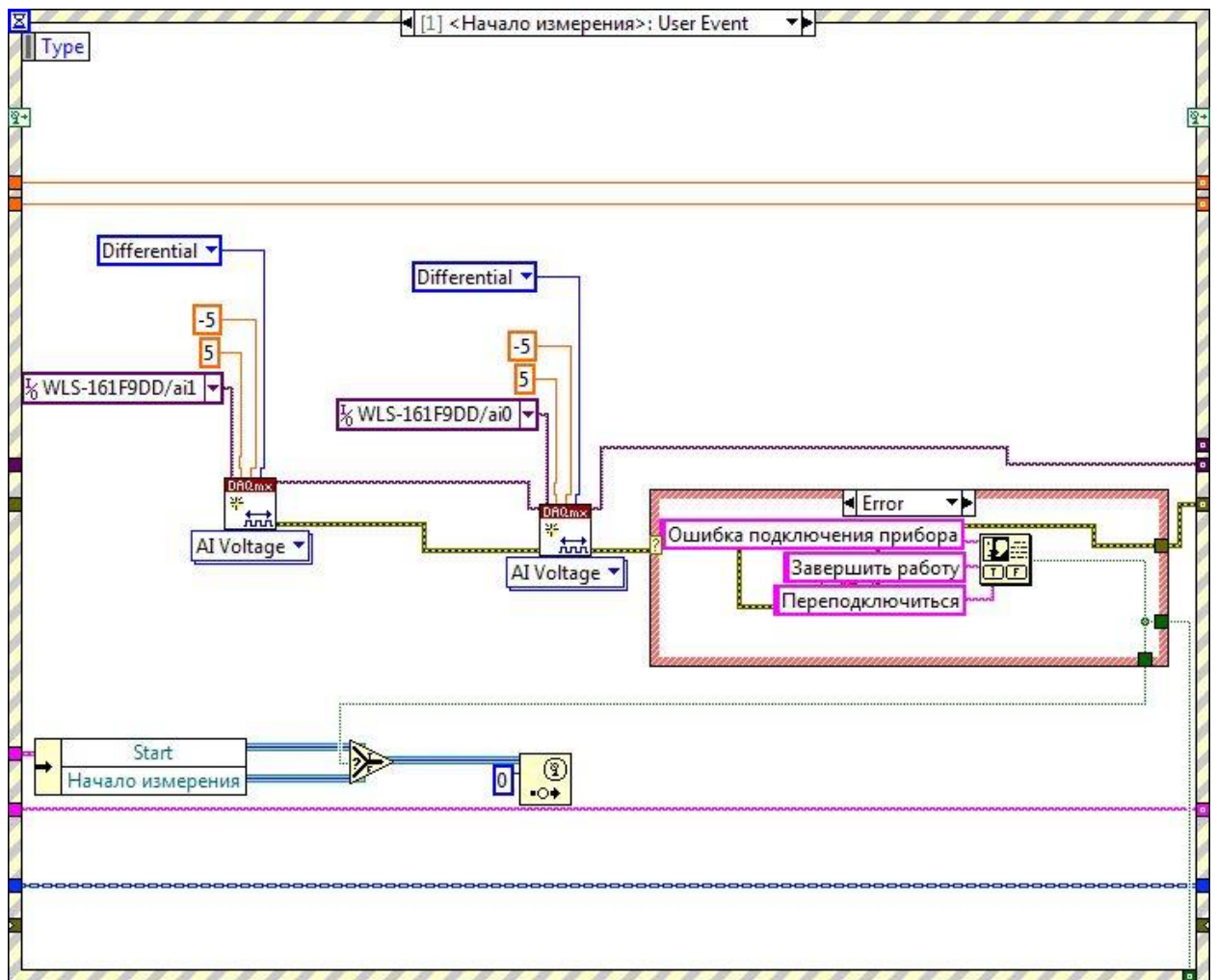
Лист

90



Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18



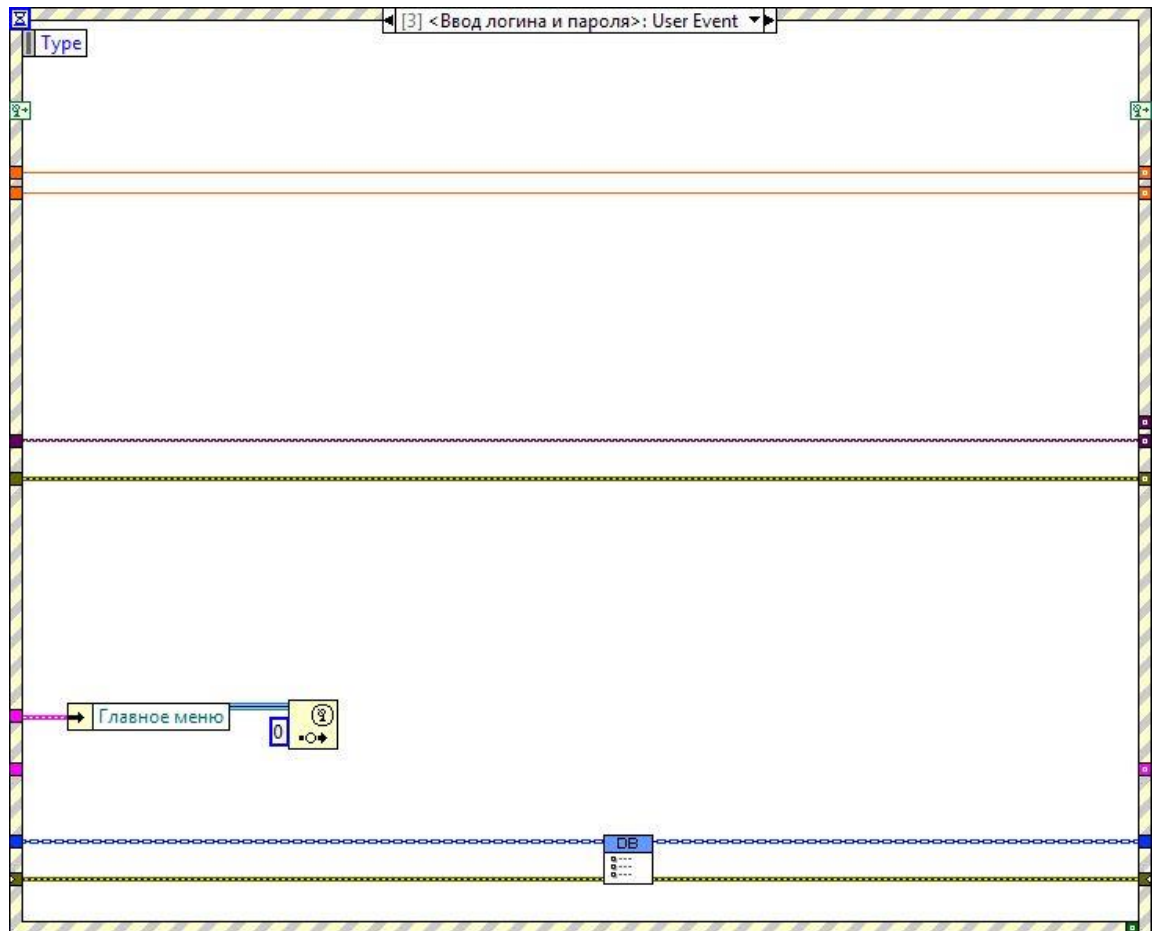
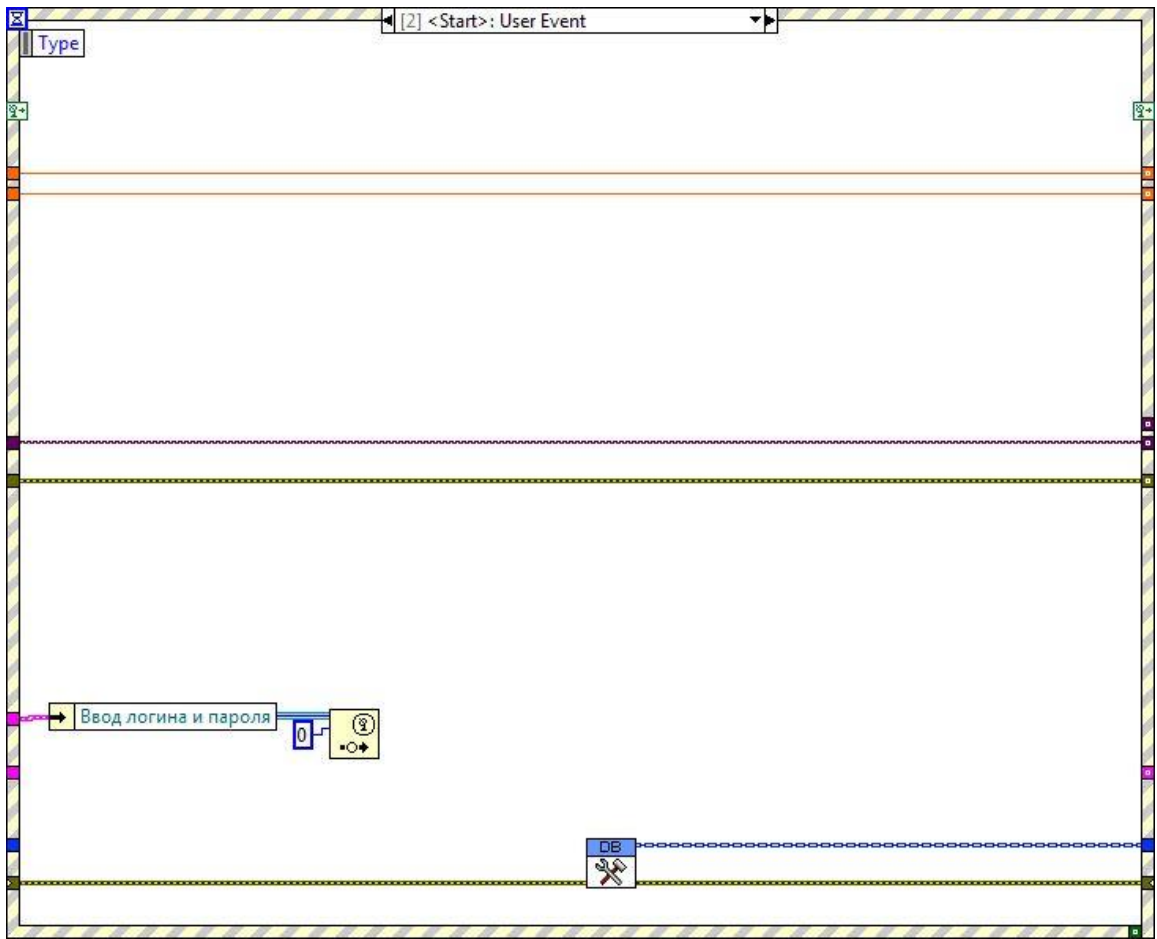
Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

92



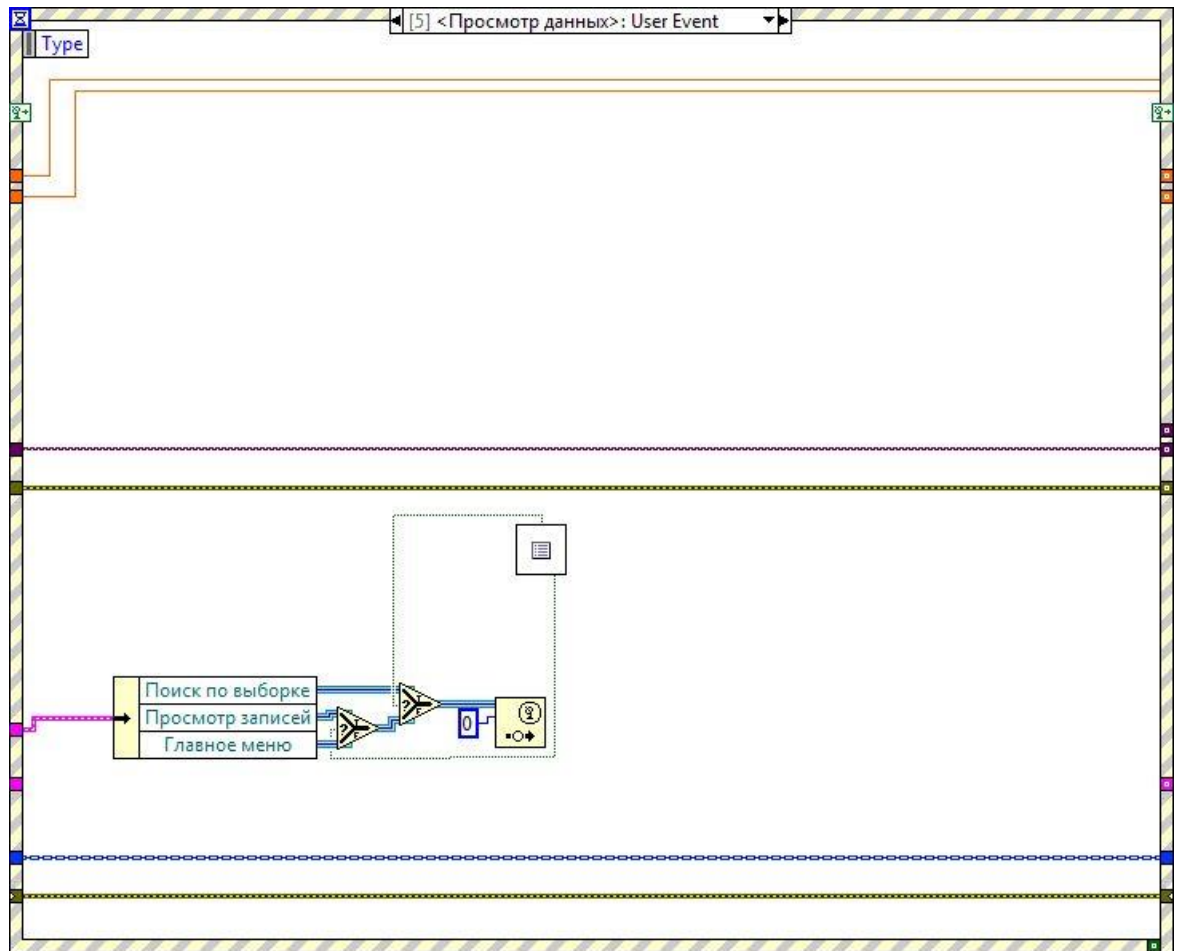
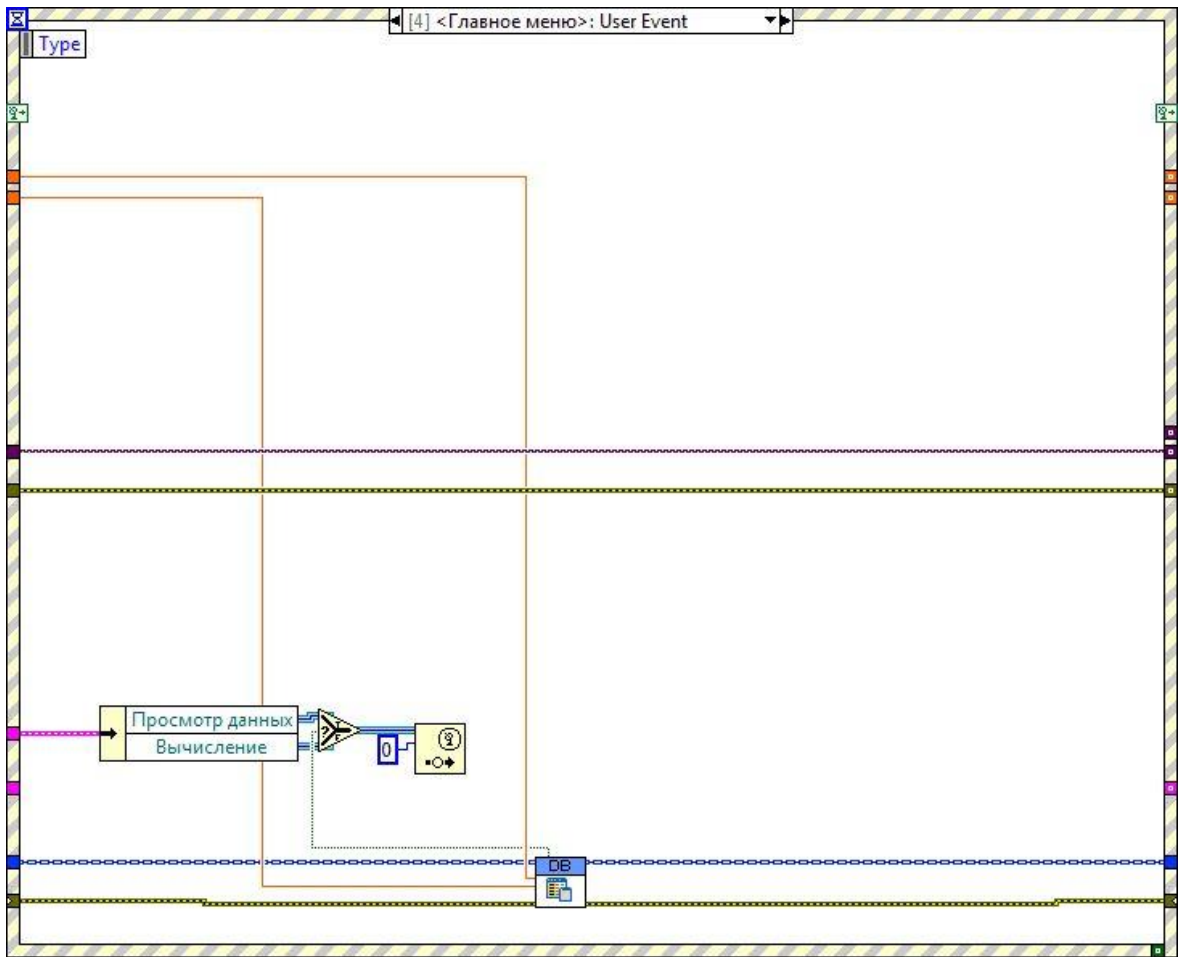


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

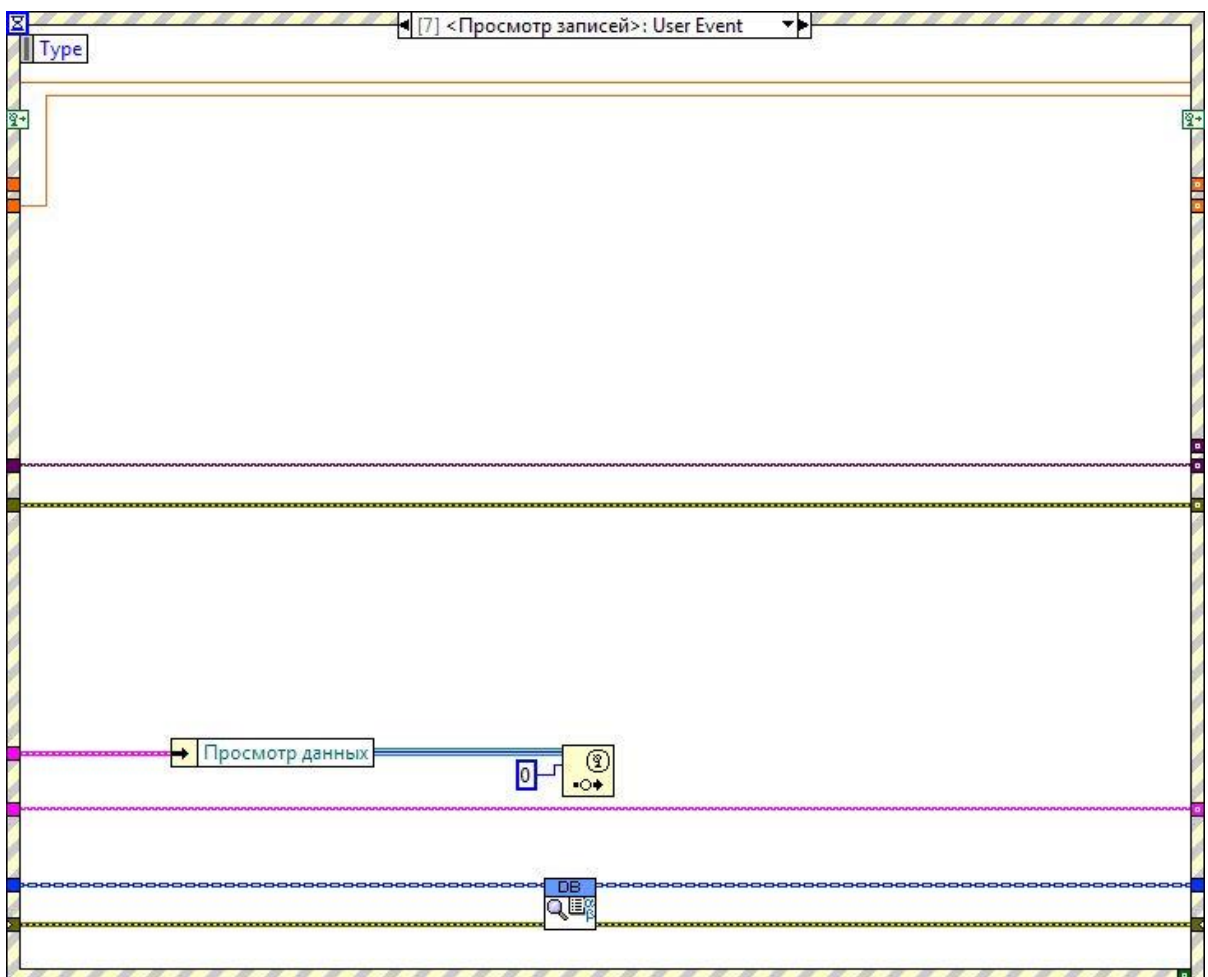
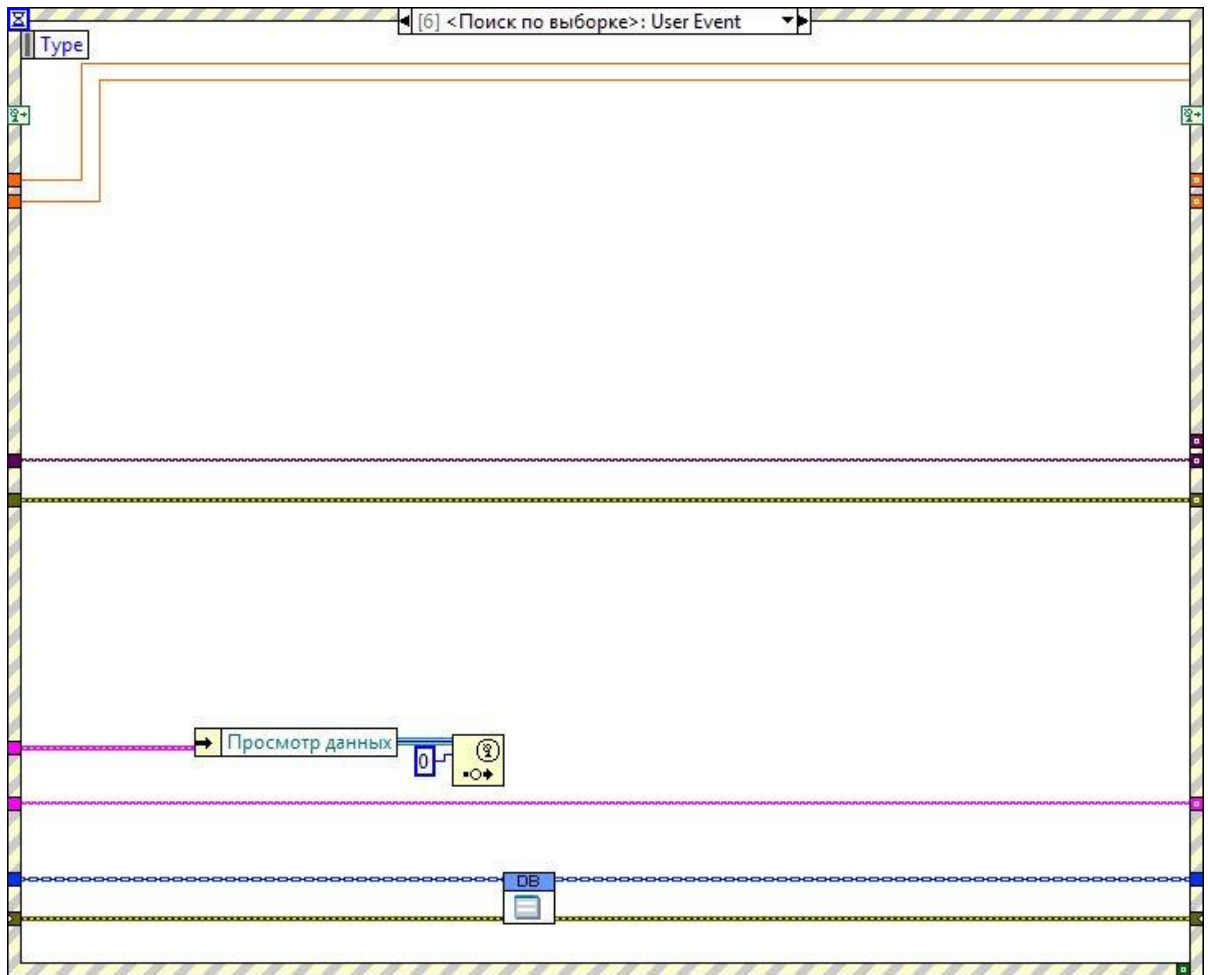
Лист

93



Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

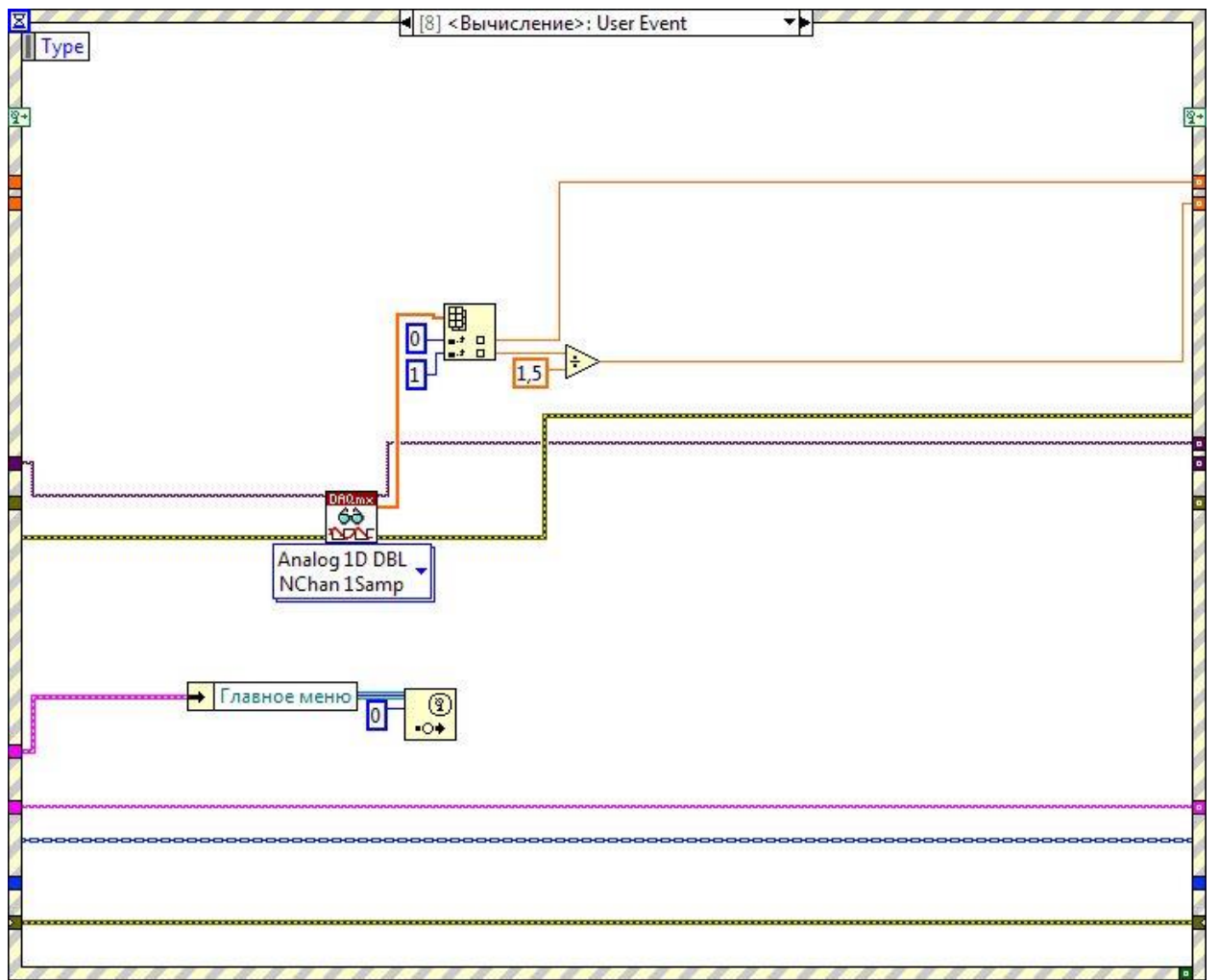


Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

95



Изм.	Лист	№ докум.	Подпись	Дата

БР-02069964-12.03.01-10-18

Лист

96

№ строки	Формат	Обозначение	Наименование	Кол.	Прим.
1					
2			<u>Документация текстовая</u>		
3					
4	A4	БР-02069964-12.03.01-10-18 ПЗ	Пояснительная записка	58	
5					
6			<u>Документация графическая</u>		
7					
8	A4	БР-02069964-12.03.01-10-18 ПЗ	Общий вид программного кода	38	Прил. А
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					

БР-02069964-12.03.01-10-18

Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Сазончик Н. Д.		20.06.18	Разработка программной части устройства для исследования параметров и характеристик цепи управления силовых тиристоров	Лист	Лист	Листов
Проб.		Ильин М. В.		20.06.18			1	1
Н. контр.		Шестеркина А. А.		20.06.18	МГУ им. Н. П. Огарёва ИЭС ЭНЭ 451 зр.			
Утв.		Беспалов Н. Н.		22.06.18				