

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
СИСТЕМ УПРАВЛЕНИЯ И СБОРА
ДАНЫХ**

*Методические указания к лабораторным работам
для студентов бакалавриата направления 15.03.04*

**САНКТ-ПЕТЕРБУРГ
2019**

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Санкт-Петербургский горный университет

Кафедра автоматизации технологических процессов
и производств

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
СИСТЕМ УПРАВЛЕНИЯ И СБОРА
ДАННЫХ

*Методические указания к лабораторным работам
для студентов бакалавриата направления 15.03.04*

САНКТ-ПЕТЕРБУРГ
2019

УДК 004.4 (073)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ УПРАВЛЕНИЯ И СБОРА ДАННЫХ: Методические указания к лабораторным работам / Санкт-Петербургский горный университет. Сост.: *А.В. Бойков, Л.Н. Никитина, А.С. Симаков* СПб, 2019. 43 с.

Содержат информацию о порядке проведения работ, об основных требованиях к содержанию отчетов. Каждая лабораторная работа содержит краткие теоретические сведения о использовании графической среды программирования LabVIEW компании «National Instruments» и описание блоков внутренних библиотек, которые могут быть использованы при выполнении задания. Ряд лабораторных работ, связанных с макетными платами и внешними устройствами, снабжены схемами электрического и сетевого подключения.

Методические указания предназначены для студентов бакалавриата направления 15.03.04 «Автоматизация технологических процессов и производств».

Научный редактор проф. *В.Ю. Бажин*

Рецензент канд. техн. наук *В.В. Васильев* (ООО «ТОМС инжиниринг»)

© Санкт-Петербургский
горный университет, 2019

ЛАБОРАТОРНАЯ РАБОТА № 1. ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЙ С ПОМОЩЬЮ ЦИКЛОВ

LabVIEW (англ. Laboratory Virtual Instrumentation Engineering Workbench) — это среда разработки и платформа для выполнения программ, созданных на графическом языке программирования «G» фирмы National Instruments (США). LabVIEW используется в системах сбора и обработки данных, а также для управления техническими объектами и технологическими процессами. Графический язык программирования «G», используемый в LabVIEW, основан на архитектуре потоков данных. Программа LabVIEW является виртуальным прибором (англ. Virtual Instrument) и состоит из двух частей:

- 1) блочной диаграммы, описывающей логику работы виртуального прибора;
- 2) лицевой панели, описывающей внешний интерфейс виртуального прибора.

В LabVIEW имеются следующие типы элементов:

- **Constant** - константа, отображается только на блок-диаграмме;
- **Control** - элемент управления;
- **Indicator** - элемент отображения данных.

Основные типы данных в LabVIEW

- 1) Числовые (**Numeric**), которые находятся в палитре с аналогичным названием **Numeric**.

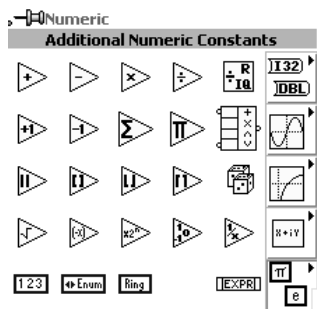


Рис. 1.1 Общий вид палитры Numeric

1.1) **Integer** – целые (обозначаются синим цветом, могут иметь следующие разряды: 8, 16, 32, 64); I – целое число со знаком; U – целое без знаковое;

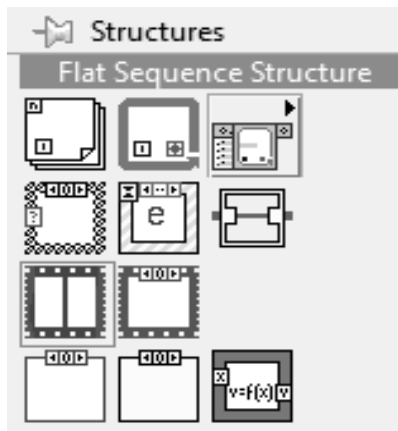





Рис. 1.4 Общий вид палитры Structures

1) **While Loop** (цикл по условию). Блок-диаграмма цикла **While** выполняется до тех пор, пока не выполнится условие выхода из цикла.



Рис 1.5 Общий вид While Loop

По умолчанию, терминал выхода из цикла имеет вид  и называется **Stop If True** (остановка, если Истина). Терминал выхода можно изменить, нажав правую кнопку мыши и выбрав пункт **Continue If True**  (продолжение, если Истина). Терминал счетчика итераций  содержит значение количества выполненных итераций. Его начальное значение всегда равно нулю.

2) **Case Structure** (структура с условным переходом). Структура **Case** имеет две или более поддиаграммы вариантов. Только одна поддиаграмма видима в данный момент времени и только одна поддиаграмма работает при выполнении данной структуры.

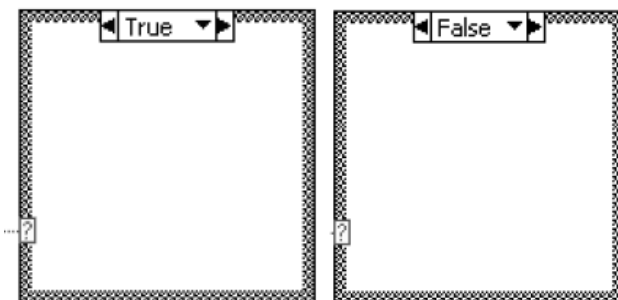

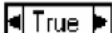




Рис. 1.6 Общий вид Case Structure

Входное значение терминала селектора структуры  определяет, какая поддиаграмма будет выполняться в данный момент. Структуру **Case** можно описать как комбинацию операторов IF, THEN, ELSE в текстовых языках программирования. Селектор структуры **Case** , состоит из указателя значения варианта и стрелок прокрутки.

3) **For Loop** (цикл с заданным числом итераций). Циклическая структура, повторяющая заключенную в нее часть кода заданное число раз. Терминал  принимает значение максимального числа итераций. Терминал счетчика итераций  содержит значение количества выполненных итераций.

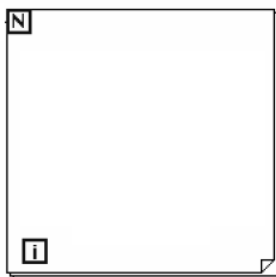


Рис. 1.7 Общий вид For Loop

Задание на лабораторную работу

В программном комплексе LabVIEW составить программу для решения кусочно-заданной системы уравнений.

Отчет по лабораторной работе должен содержать:

- 1) Описание хода выполнения.
- 2) Скриншот алгоритма программы и лицевой панели.

Рекомендация: Основные горячие клавиши, которые будут полезны при работе с LabVIEW представлены ниже:

<Ctrl+R> запуск виртуального прибора (ВП);

<Ctrl+E> переключение между лицевой панелью и блок-диаграммой;

<Ctrl+H> вывод на экран/скрытие окна контекстной подсказки;

<Ctrl+B> удаление всех разорванных проводников данных;

<Ctrl+F> поиск объектов, находящихся в памяти ВП.

ЛАБОРАТОРНАЯ РАБОТА № 2. РАБОТА СО СТРОКОВЫМИ ФУНКЦИЯМИ

Кроме числовых и логических типов данных, рассмотренных в первой лабораторной работе, в LabVIEW есть строковый тип данных - **String**. Работа с текстовыми функциями осуществляется через соответствующую палитру **String**

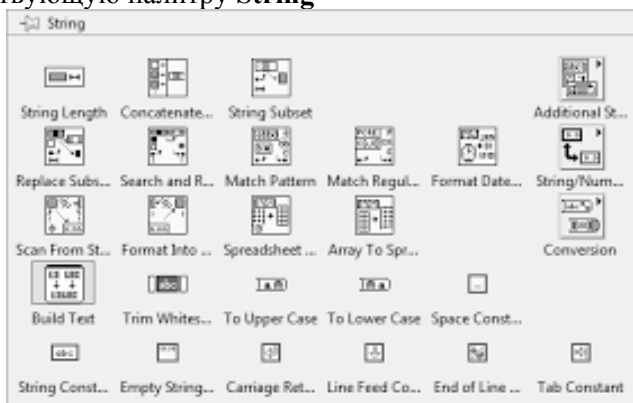


Рис. 2.1 Общий вид палитры Strings

С описанием функций данной палитры можно ознакомиться через контекстную подсказку <Ctrl+H>. Среди функций можно выделить следующие: **String Length** (определение длины строки в символах), **Concatenate String** (объединение двух и более строк), **Search and replace String** (поиск в пределах строки и замена) и другие.

Задание на лабораторную работу

В программном комплексе LabVIEW составить программу выполняющую следующие функции:


- 1) Генерация массива дробных чисел из **N** элементов, случайно изменяющихся от **MIN** до **MAX**.
- 2) Подсчет количества элементов массива больше **X**.
- 3) Для непрерывной работы программы воспользоваться циклом **While Loop**. Автоматический выводить результаты расчета в виде текстовой строки (шаблон строки имеет вид: «В массиве из **%N** элементов, изменяющихся от **%MIN** до **%MAX**, содержится **%K** чисел больше **%X**»). Элементы шаблона, выделенные условным

знаком «%», заменяя при каждом расчете с помощью функций палитры String.

Отчет по лабораторной работе должен содержать:

- 1) Описание хода выполнения.
- 2) Скриншот алгоритма программы и лицевой панели.

Рекомендация: Для реализации счетчика можно (но не обязательно) использовать сдвиговые регистры.

Сдвиговые регистры  используются при работе с циклами для передачи значений от текущей итерации цикла к следующей и аналогичны статическим переменным в текстовых языках программирования.

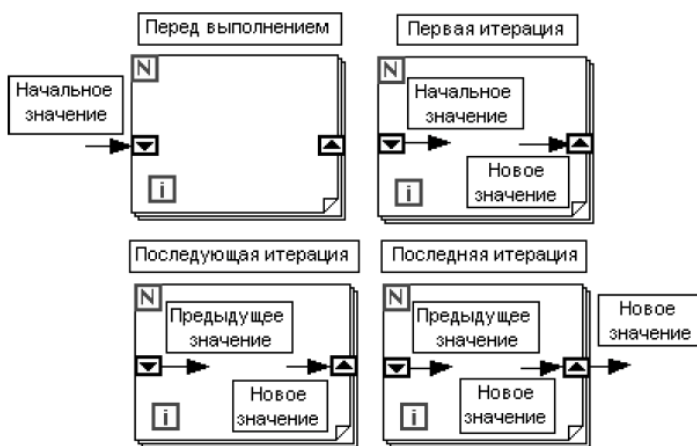


Рис. 2.2 Принцип работы сдвиговых регистров

Сдвиговой регистр создается щелчком правой кнопки мыши по границе цикла и выбором из контекстного меню пункта **Add Shift Register**.

ЛАБОРАТОРНАЯ РАБОТА № 3. ИЗМЕРЕНИЕ И АНАЛИЗ ДАННЫХ ПО КАНАЛУ ТЕМПЕРАТУРА

С помощью подключения модуля расширения и программы конфигурирования устройств Measurement & Automation Explorer (NI MAX) можно осуществлять сбор и управления данными. Утилита NI MAX определяет все аппаратные средства National Instruments.

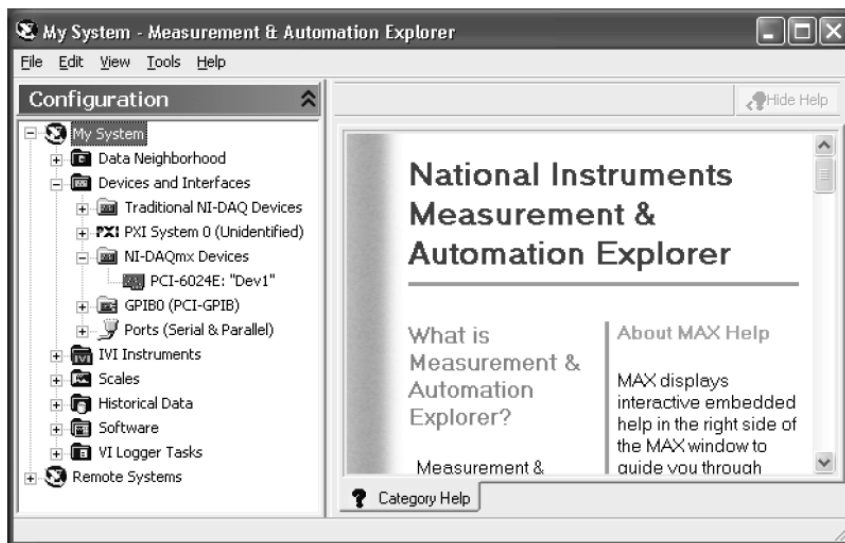


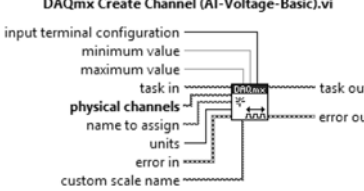
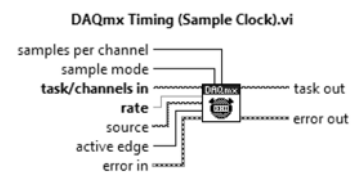
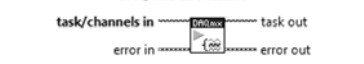
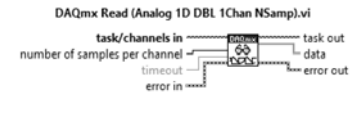

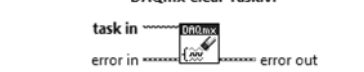
Рис. 3.1 Начальное окно конфигурационной утилиты NI MAX

Измерение температуры проводится при помощи терморезистора (является параметрическим датчиком, который меняет свои параметры под действием измеряемой величины), находящегося в подключаемом модуле платы расширения. В списке устройств, определяемых NI MAX, в блоке NI-DAQmx Devices следует найти устройство **PXI-6251** и запомнить его обозначение (например, “Dev3»).

Подключение к устройству PXI-6251 осуществляется через палитру Measurement I/O (Измерение) –NI-DAQmx – Data Acquisition (NI-DAQ - набор драйверов для оборудования National Instruments). В палитре NI-DAQmx – следует найти и выбрать следующие функции:

Таблица 3.1

Описание основных блоков палитры Data Acquisition

Внешний вид функции	Описание и назначение
<p style="text-align: center;">DAQmx Create Channel (AI-Voltage-Basic).vi</p> 	<p>Создание задачи (виртуального канала). Требуется обязательно подключить physical channels (физический канал) согласно обозначению устройства, определенного в NI MAX и номеру канала модуля расширения, к которому подключен терморезистор. Также требуется настроить тип измеряемого параметра и его пределы варьирования.</p>
<p style="text-align: center;">DAQmx Timing (Sample Clock).vi</p> 	<p>Настройка частоты измерения (опроса датчика). Среди обязательных параметров: rate – количество измерений в секунду, sample mode – режим измерения (finite samples – однократное измерение, continuous samples – непрерывное измерение)</p>
<p style="text-align: center;">DAQmx Start Task.vi</p> 	<p>Запуск измерений</p>
<p style="text-align: center;">DAQmx Read (Analog 1D DBL 1Chan NSamp).vi</p> 	<p>Считывание данных из буфера. Среди входных параметров: number of samples per channel – количество считываемых параметров, timeout – время ожидания данных.</p>
<p style="text-align: center;">DAQmx Stop Task.vi</p> 	<p>Остановка измерений</p>
<p style="text-align: center;">DAQmx Clear Task.vi</p> 	<p>Освобождает занятые задачей ресурсы</p>

Задание на лабораторную работу

Создать программу сбора данных с терморезистора в среде LabVIEW, используя блоки палитры NI-DAQmx, а также структуру While Loop для непрерывного измерения показаний температуры.

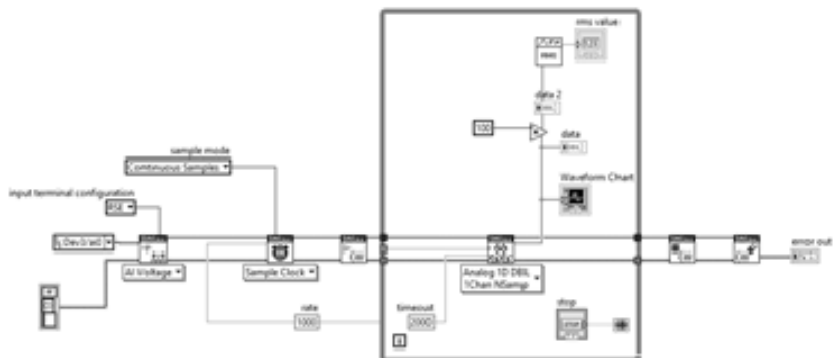


Рис. 3.2 – Пример блок-диаграммы

Вывести результаты на Front Panel в виде числового индикатора и графика Waveform Chart. Отображаемые данные предварительно обработать с помощью среднеарифметического усреднения данных **RMS**, масштабировать измеренный сигнал для перевода размерности считываемых данных из мВ в градусы Цельсия.

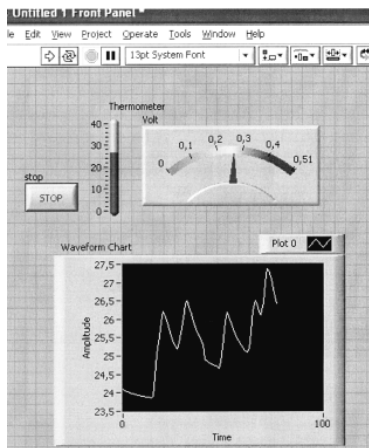


Рис. 3.3 Пример лицевой панели

Отчет по лабораторной работе должен содержать:

- 1) Описание хода выполнения.
- 2) Скриншот алгоритма программы и лицевой панели.

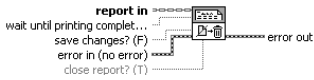
ЛАБОРАТОРНАЯ РАБОТА № 4. СОЗДАНИЕ ОТЧЕТНЫХ ФОРМ ДЛЯ MS WORD

В программном пакете LabVIEW имеются средства для создания отчетов в форматах doc, xls, html. Основные блоки палитры Report Generation приведены в таблице.

Таблица 4.1

Описание основных блоков палитры Data Acquisition

Внешний вид функции	Описание и назначение
<p style="text-align: center;">New Report</p>	<p>Создание нового отчета с указанием типа report type (Windows - стандартный отчет (по умолчанию), HTML - отчет в формате HTML, Word - отчет в MS Word, Excel - отчет в MS Excel) и формирование ссылки на него report out для последующего взаимодействия с отчетом.</p>
<p style="text-align: center;">Append Report Text</p>	<p>Добавление Текста в текущий отчет.</p>
<p style="text-align: center;">Append Table to Report</p>	<p>Добавление Таблицы в текущий отчет.</p>
<p style="text-align: center;">Append List to Report</p>	<p>Добавление Списка в текущий отчет.</p>
<p style="text-align: center;">Append Image to Report</p>	<p>Добавление Картинки в текущий отчет с указанием пути к выбранному изображению - path or URL of image.</p>
<p style="text-align: center;">Save Report to File</p>	<p>Сохраняет текущий отчет в файл.</p>

Внешний вид функции	Описание и назначение
<p style="text-align: center;">Dispose Report</p> 	<p>Закрывает файл отчета и закрывает интерфейс.</p>

Задание на лабораторную работу

Создать отчет в формате Word средствами LabVIEW.

Отчет по лабораторной работе должен содержать титульный лист, пример текста, картинки, списка и картинки:

- 1) Описание хода выполнения.
- 2) Скриншот алгоритма программы и лицевой панели.

ЛАБОРАТОРНАЯ РАБОТА № 5. КОНФИГУРИРОВАНИЕ СИСТЕМЫ COMPACT FIELDPOINT

Освоить навыки подключения и настройки используемого в промышленности оборудования, создания системы регулирования в объектно-ориентированной среде, а также проверить знания в области автоматике.

Compact FieldPoint представляет собой программируемый контроллер автоматизации, разработанный для решения задач автономного промышленного управления, сбора данных и их передачи по сети. Входящая в семейство объединительная панель отличается повышенной механической прочностью. Винтовой механизм крепления, позволяет FieldPoint-продуктам благополучно переносить высокие вибрационные нагрузки (до 5 g), встречающиеся в промышленности и транспортных средствах. При размерах 5x10x5 дюймов модули выдерживают удары силой до 50 g и способны работать в температурном диапазоне от -25°C до 60°C. Семейство FieldPoint дает возможность использовать встраиваемые LabVIEW-приложения в очень жестких условиях, не приемлемых для многих персональных компьютеров промышленного класса.

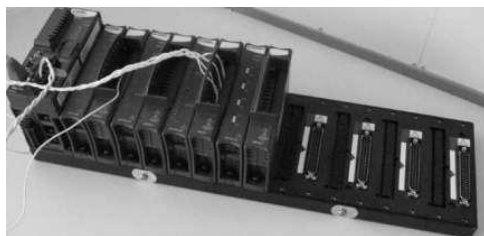


Рис. 5.1 Внешний вид контроллера Compact FieldPoint

В лабораторной работе используется микроконтроллер cFP-2020. К контроллеру подключены следующие модули:

- 1) cFP-TC-120 являются модулем с 8 входными каналами для прямого измерения температуры термодпар стандарта J, K, T, N, R, S, E, V.
- 2) cFP-AIO-610 - комбинаторный аналоговый модуль с 8 каналами ввода-вывода.

В качестве объекта управления используется лабораторный блок «Термостат».

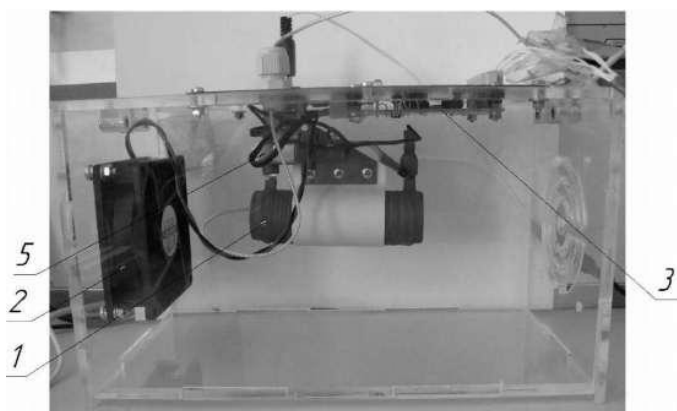
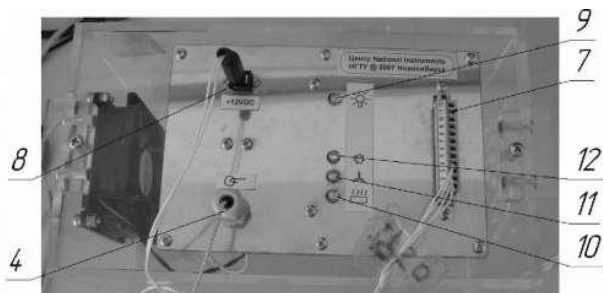


Рис. 5.2 Лабораторный блок «Термостат»

Блок «Термостат» состоит из следующих элементов:

1. Нагревательный элемент.
2. Вентилятор с двигателем постоянного тока.
3. Монтажная плата с силовыми ключами управления нагревателем и двигателем вентилятора (закрыта металлической защитной панелью).
4. Термопара с соединительным кабелем (рабочий конец термопары размещен внутри нагревательного элемента, выноска указывает на кабельный ввод, через который выводится кабель подключения термопары).
5. Температурное реле защитного отключения.
6. Монтажная плата с датчиком скорости вращения.
7. Разъем с клеммными зажимами для подключения цепей

управления (X2).

8. Разъем для подключения блока питания (XI).
9. Индикатор включения питания.
10. Индикатор включения нагревателя.
11. Индикатор включения вентилятора.
12. Индикатор резервный.

Задание на лабораторную работу

Перед началом работы необходимо подключить стенд к микроконтроллеру. Настроить программное обеспечение на ПК. Необходимо быть внимательным, выполняя эту процедуру, так как в противном случае работа контроллера и стенда будет некорректной.

Составить программу в программе LabVIEW. Рекомендуется выполнять программу по следующему алгоритму (неординарные решения крайне приветствуются, если они приведут к поставленной цели).

Порядок подключения лабораторной установки к микроконтроллеру

По заданию необходимо собрать стенд для регулирования температуры в Термостате с помощью вентилятора и нагревательного элемента. Данный стенд включает следующие компоненты:

- Термостат;
- Блок питания термостата;
- Контроллер FieldPoint с модулями: DIO-550 (дискретный), TC-120 (для подключения термопар);
- Блок питания контроллера;
- Персональный компьютер с программным обеспечением LabVIEW;
- Кабель для соединения контроллера с термостатом.

Подключение электрической и коммуникационной сетей.

Подключение вентилятора и нагревательного элемента производится к дискретному модулю DIO-550. Необходимо соединить клеммы термостата и контроллера следующим образом: термопару необходимо подключить к модулю TC-120 (клеммы 1, 2).

Подключить блок питания к термостату, используя соответствующие разъемы и блок питания к контроллеру.

Подключить контроллер с помощью сетевого кабеля к персональному компьютеру.

Подключить блоки питания к сети.

Настройка микроконтроллера

– Запустить приложение отладки «Measurement & Automation».

– Проверить наличие подключения контроллера (пункт «Remote System»). Если подключение не обнаружено, необходимо сделать обновление с помощью кнопки «Refresh» (или клавиша F5).

Часть 1. Дискретное регулирование стэнда

Запустите приложение LabVIEW и создайте в нем Real Time проект на базе контроллера Field Point. Выберите пункт New Blank VI. В появившемся окне Front panel соберите панель управления стэндом.

Соберите блок-диаграмму и лицевую панель согласно примеру.

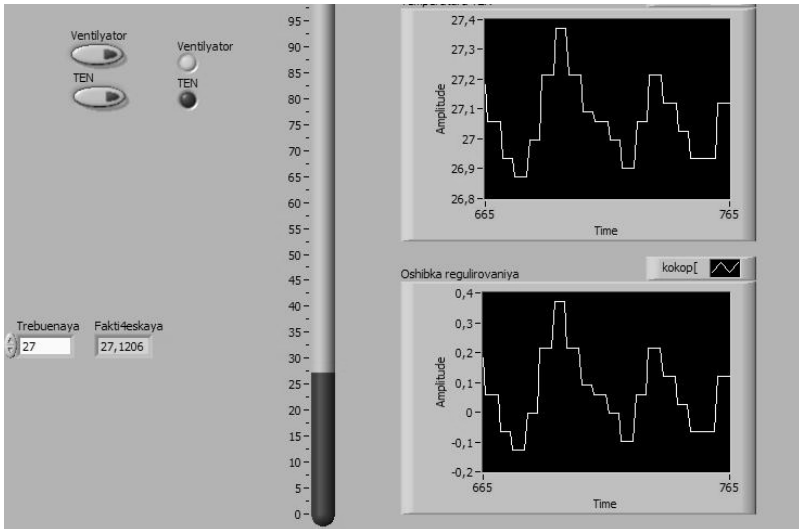


Рис. 5.3 Пример лицевой панели для режима дискретного регулирования

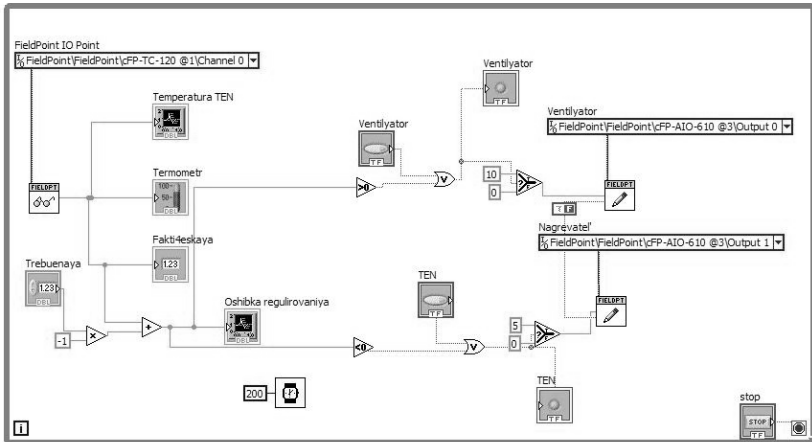


Рис. 5.4 Пример блок-диаграммы для режима дискретного регулирования

Часть 2. ШИМ-регулирование вентилятора

Выберите пункт New Blank VI. В появившемся окне Front Panel соберите панель управления стандом. Соберите блок-диаграмму и лицевую панель согласно примеру.

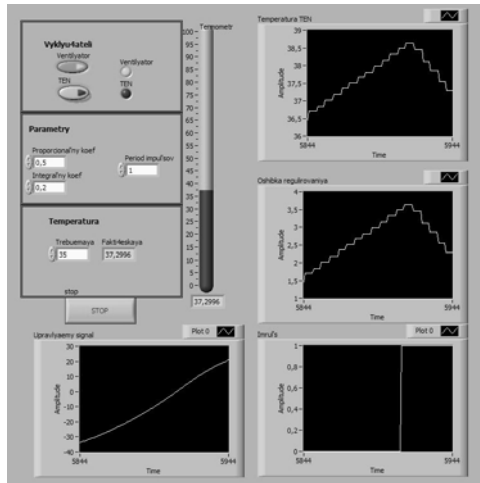


Рис. 5.5 Пример лицевой панели для режима ШИМ-регулирования вентилятора

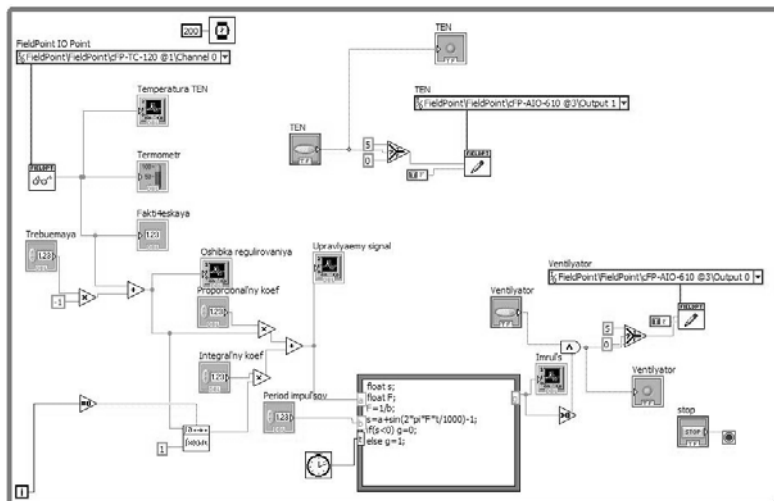


Рисунок 5.6 Пример блок-диаграммы для режима ШИМ-регулирования вентилятора

Рекомендация: Расчет параметров ШИМ осуществляется через Formula Node. Нажмите на рамку элемента Formula Node и создайте три входа. Назовите их «a», «b», «t». Создайте один выход и назовите его «g».

Внутри рамки необходимо записать:

```
float s;
float F;
F=1/b;
s=a+sin(2*pi*F*t/1000)-1;
if(s<0) g=0; else g=1;
```

Часть 3. ШИМ регулирования ТЭНа

Выберите пункт New Blank VI. В появившемся окне Front panel соберите панель управления стендом. Соберите блок-диаграмму и лицевую панель согласно примеру.

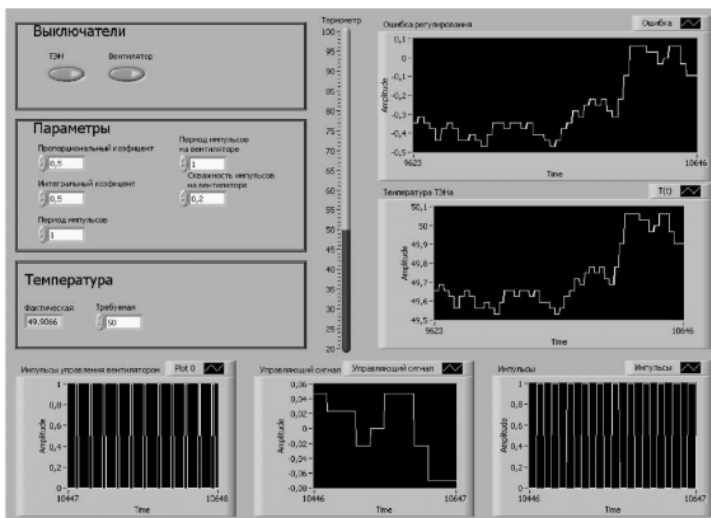


Рис. 5.7 Пример лицевой панели для режима ШИМ-регулирования ТЭНа

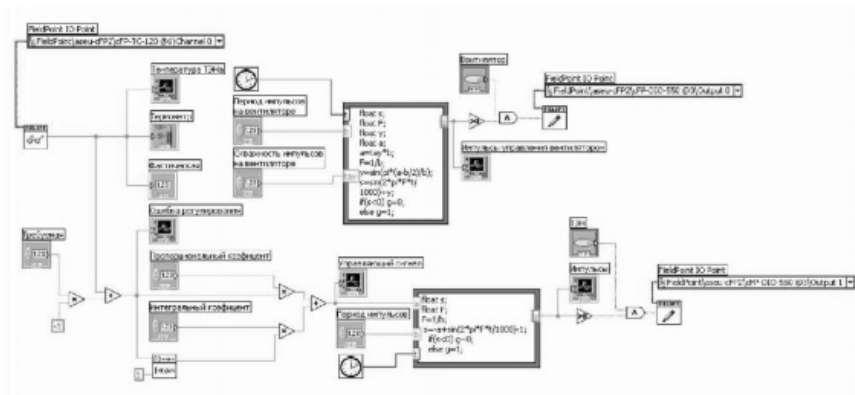


Рис. 5.8 Пример блок-диаграммы для режима ШИМ-регулирования ТЭНа

Рекомендация: Расчет параметров ШИМ осуществляется через Formula Node.

Внутри первой рамки Formula Node необходимо записать:

float s;

float F;

$$F=1/b; s=a+\sin(2*\pi*F*t/1000)-1;$$

if(s<0) g=0; else g=1;

Внутри второй рамки Formula Node2 необходимо записать:

float s;

float F;

float y;

float a;

*a=tay*b1;*

F=1/b1y=sin(pi(a-b1/2)/b1);*

*s=sin(2*pi*F*t/1000)+y;*

if (s<0) g1=0;

else g1=1;

Отчет по лабораторной работе должен содержать:

- 1) Описание хода выполнения.
- 2) Скриншот алгоритма программы и лицевой панели.
- 3) Выводы по работе различных режимов регулирования.

ЛАБОРАТОРНАЯ РАБОТА № 6. КОНФИГУРИРОВАНИЕ ПЛК COMPACTRIO

Разработка приложения для CompactRIO (Reconfigurable Embedded System) в среде LabVIEW по созданию простейшего термостата. Знакомство с основными принципами и методами программирования в среде LabVIEW на платформе CompactRIO.

Для выполнения задания необходимо следующее оборудование:

- встраиваемый контроллер реального времени cRIO-9012
- шасси расширения – cRIO 9104;
- модуль измерения температуры - NI 9401;
- модуль цифрового ввода/вывода - NI 9211;
- источник питания;
- кабель для подключения к сети Ethernet;
- учебный стенд – «Термостат».

Задание на лабораторную работу

Настройка оборудования

Чтобы подготовить оборудование к работе, выполните следующие действия:

1. Установите контроллер cRIO-9012 на шасси cRIO-910x.
2. Установите модуль ввода/вывода NI 9401 в слот 1 шасси и модуль NI 9211 в слот 2.
3. Подключите термопару к каналу 0 модуля NI 9211. Подробнее см. инструкцию по эксплуатации модуля NI 9211.
4. Канал 0 модуля термодатчика будем использовать для считывания показаний температуры.
5. Канал 16 модуля цифрового вывода будем использовать для управления вентилятором (DIO 1). Канал 14 — для управления нагревателем (DIO 0).
6. Подключите источник питания к контроллеру и установите Ethernet-соединение в соответствии с инструкцией по эксплуатации CompactRIO cRIO-9012.
7. Изменить IP адрес сетевого подключения.

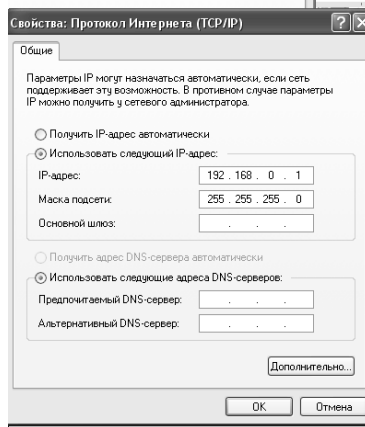


Рис. 6.1 Сетевые настройки CompactRIO

Часть 1. CompactRIO в режиме Scan Mode

Для организации и управления виртуальными приборами и целевыми устройствами приложения используется LabVIEW Project Explorer.

Выполните следующие действия, чтобы создать проект LabVIEW для реконфигурируемой встраиваемой системы CompactRIO и добавить в него модули ввода/вывода:

1. Запустите LabVIEW.
2. Выберите NEW, затем пункт Real-Time Project, чтобы открыть окно Project Explorer. Задайте любое имя проекта.

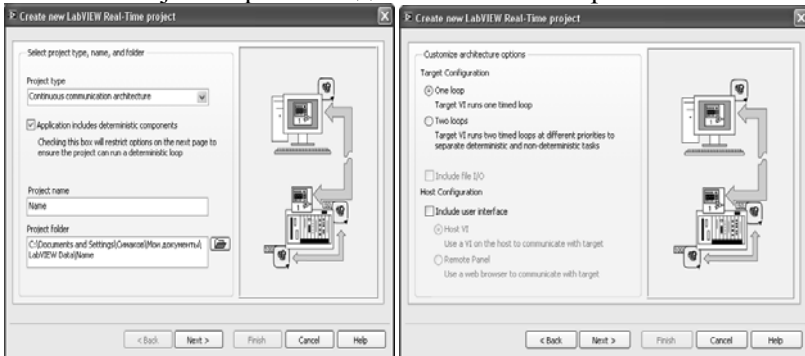


Рис. 6.2 Настройки проекта

3. Проект будет создан на базе сконфигурированного оборудования, поэтому выберите пункт Existing target, откроется список доступных целевых устройств. CompactRIO будет работать в режиме Scan Interface

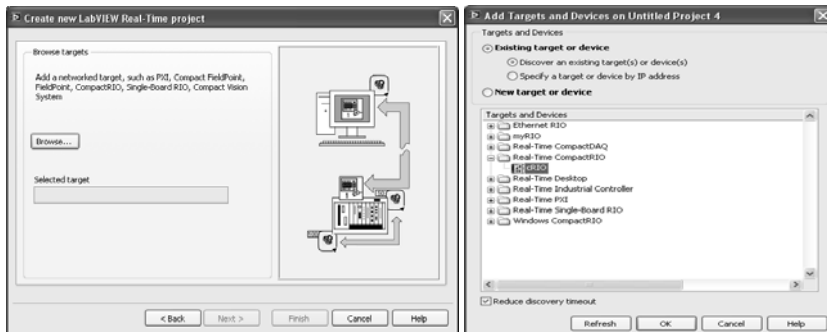


Рис. 6.3 Настройки подключаемого оборудования

4. Настройте модули цифрового ввода-вывод и модуль термодпар следующим образом:

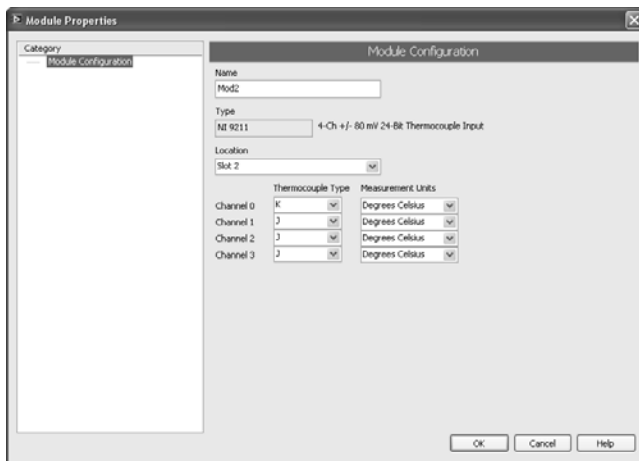


Рис. 6.4 Настройка модуля термодпарного ввода

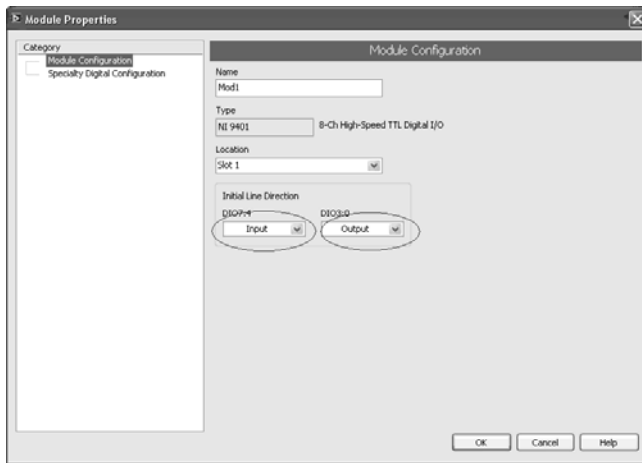


Рис. 6.5 Настройка модуля цифрового вывода

5. Создайте новый VI файл для CompactRIO и воссоздайте в нём модель.

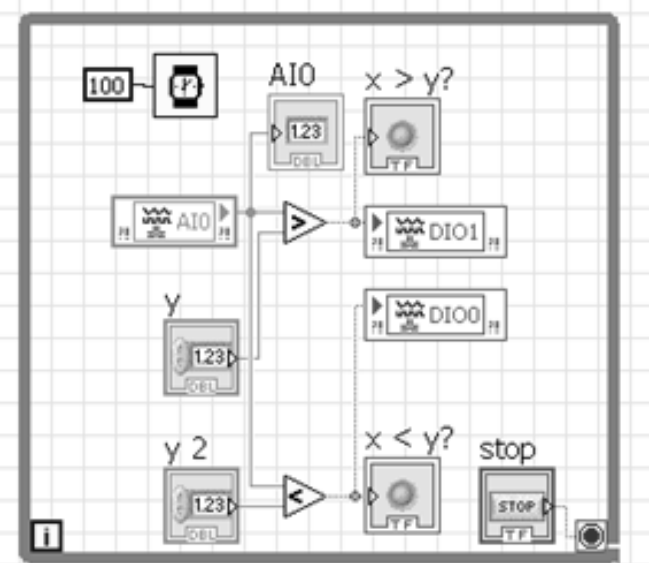


Рис. 6.6 Блок-схема дискретного регулирования CompactRIO

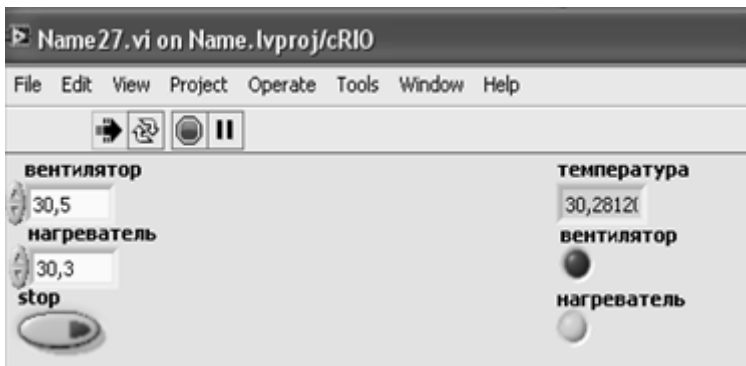


Рис. 6.7 Лицевая панель дискретного регулирования CompactRIO

Часть 2. CompactRIO в режиме FPGA Interface

Для организации и управления виртуальными приборами и целевыми устройствами приложения используется LabVIEW Project Explorer.

Выполните следующие действия, чтобы создать проект LabVIEW для реконфигурируемой встраиваемой системы CompactRIO и добавить в него модули ввода/вывода:

1. Запустите LabVIEW.
2. Выберите NEW, затем пункт Real-Time Project, чтобы открыть окно Project Explorer. Задайте любое имя проекта.

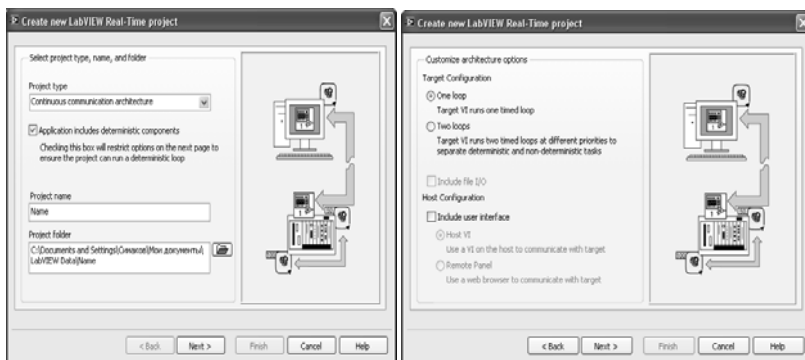


Рис. 6.8 Настройки проекта

3. Проект будет создан на базе сконфигурированного оборудования, поэтому выберите пункт Existing target, откроется список доступных целевых устройств. CompactRIO будет работать в режиме LabVIEW FPGA Interface

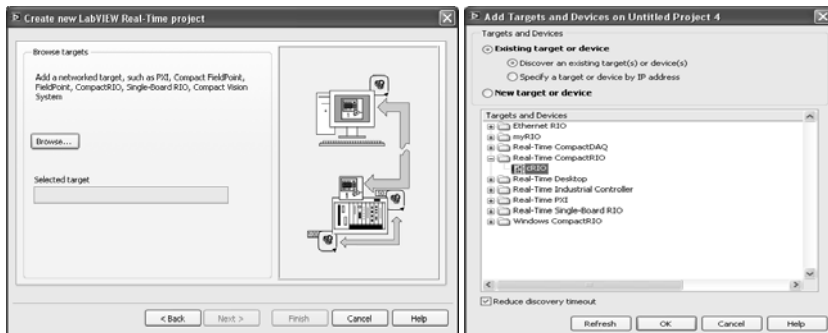


Рис. 6.9 Настройки подключаемого оборудования

4. Настройте модули цифрового ввода-вывод и модуль термодатчиков следующим образом:

Кликните по названию канала **TC0** в столбце **Mod2**, во вкладке **General** измените имя на actual temperature - RAW.

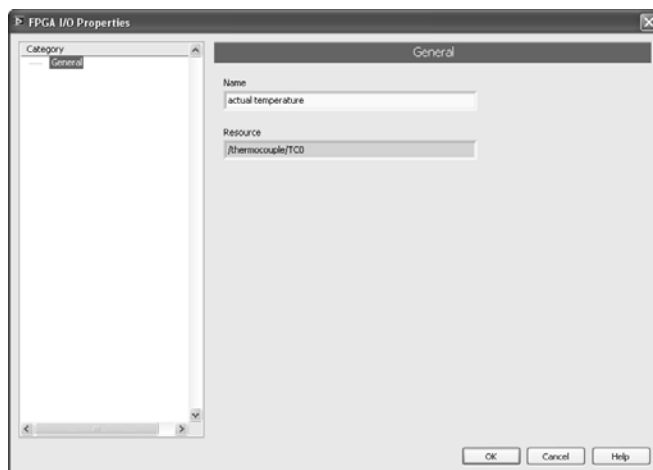


Рис. 6.10 Настройка модуля термодатчиков

1. Кликните на имя канала **DIO0** в столбце **Mod 1**, во вкладке

General измените имя на heater control.

2. Кликните на имя канала **DIO1** в столбце **Mod 1**, во вкладке **General** измените имя на fan control.

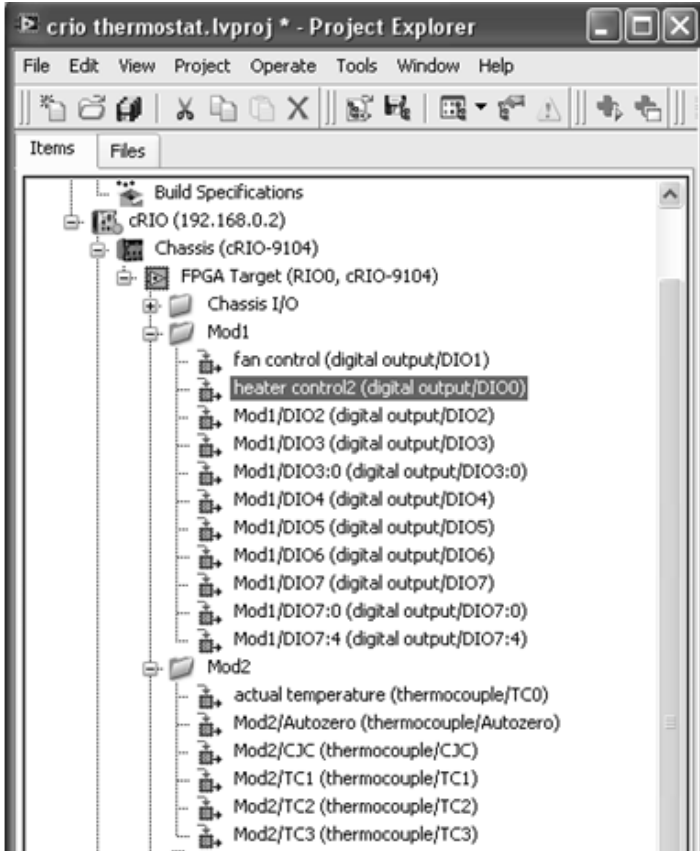


Рис. 6.11 *Настройка модуля цифрового вывода*

5. Чтобы создать FPGA VI, необходимо создать новый специальный виртуальный прибор для целевого устройства FPGA, затем скомпилировать его и загрузить в целевое устройство.

Чтобы добавить в проект виртуальный прибор для нужного целевого устройства FPGA, выполните следующие действия:

1. В окне **Project Explorer** кликните правой кнопкой мыши по

названию целевого устройства FPGA и выберите **New»VI**. Появится окно нового виртуального прибора.

2. В главном меню лицевой панели или блок-диаграммы выберите **File»Save As**, откроется диалоговое окно **Name the VI**.

3. Введите `crio thermostat (fpga)` в поле **File name**.

4. Нажмите **OK**.

Считывание показаний температуры

Выполните следующие действия для создания ВП `crio thermostat (fpga) VI`.

1. Поместите **While Loop** на блок-диаграмму `crio thermostat (fpga) VI`.

2. Кликните правой кнопкой мыши по терминалу условия выхода и выберите в меню **Create Control**.

3. Откройте палитру функций и поместите узел **FPGA I/O Node** в цикл **While Loop**.

4. Кликните правой кнопкой мыши по узлу **FPGA I/O Node** и выберите **Select FPGA I/O»Analog Input» thermocouple input»actual temperature**.

5. Кликните правой кнопкой мыши по выходу **actual temperature** узла **FPGA I/O Node** и выберите **Create»Indicator**.

6. Назовите индикатор **actual temperature**.

Управление температурой

Чтобы осуществлять управление температуры, **FPGA VI** должен сравнивать текущую температуру с заданным диапазоном и определять, когда нужно включить вентилятор или нагреватель.

Выполните следующие действия, чтобы добавить в ВП возможность сравнения показаний температур и управления работой вентилятора и нагревателя:

1. Поместите два объекта **numeric control** (числовой элемент управления) на лицевую панель. Назовите их **low target temperature** и **high target temperature**.

Элемент управления **low target temperature** будет задавать нижний предел температурного диапазона, а **high target temperature** – верхний.

2. Поместите узлы **Less?** (меньше?) и **Greater?** (больше?) в

цикл While loop на блок-диаграмме.

3. Соедините выход **actual temperature** узла FPGA I/O Node с входом **x** функций Less? и Greater?

4. Соедините выход **low target temperature** с входом **y** функции Less? Соедините выход **high target temperature** с входом **y** функции Greater?

5. Поместите еще один узел FPGA I/O Node в цикл While Loop справа от первого FPGA I/O Node.

6. Кликните правой кнопкой мыши по новому узлу FPGA I/O Node и выберите **Select FPGA I/O»Digital Line Output»digital output»fan control**.

7. Кликните правой кнопкой мыши по узлу FPGA I/O Node и выберите **Add Element**.

8. Кликните правой кнопкой по новому элементу FPGA I/O Node и выберите **Select FPGA I/O»Digital Line Output»digital output»heater control**.

9. Кликните правой кнопкой мыши по элементу **fan control** и выберите **Change to Write**.

10. Соедините выход **x<y?** функции Less? с входом **heater control** второго узла FPGA

I/O Node. Соедините выход **x>y?** функции Greater? с входом **fan control**.

11. Соедините выход **Error Out** первого узла FPGA I/O Node с входом **Error In** второго узла FPGA I/O Node.

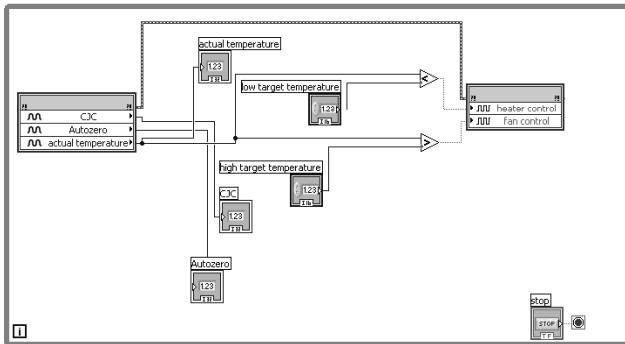


Рис. 6.12 Блок-схема дискретного регулирования CompactRIO

Компиляция и загрузка FPGA VI

Выполните следующие действия для компиляции и загрузки FPGA VI в целевое устройство FPGA.

1. Кликните правой кнопкой мыши в окне **Project Explorer** по названию целевого устройства FPGA (реконфигурируемое шасси), и выберите **Properties**. Появится диалоговое окно **FPGA Target Properties**.

2. Отметьте пункт **Run when loaded to FPGA**.

3. Выберите раздел **Top-Level Clock (максимальная частота)** в левой части диалогового окна **FPGA Target Properties**.

4. Убедитесь, что выбрана default clock (частота по умолчанию).

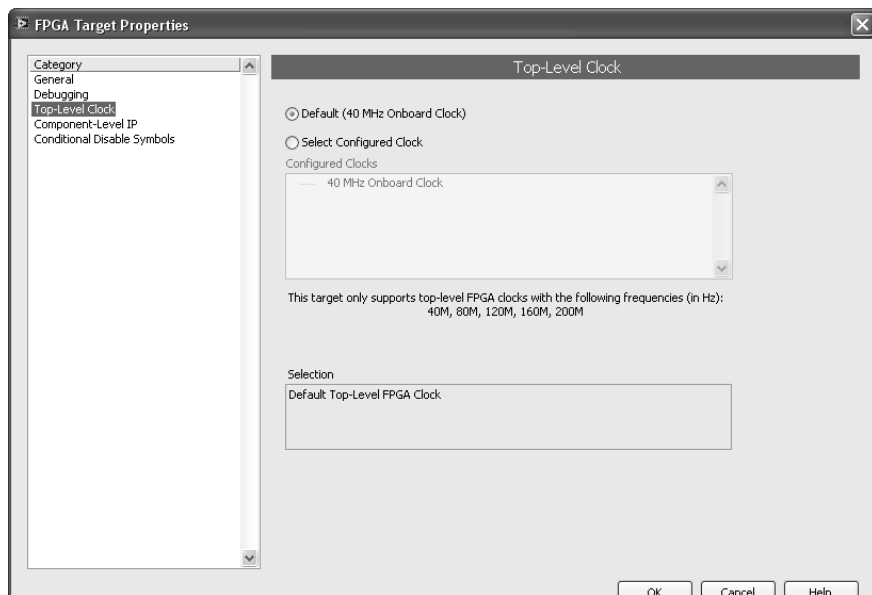


Рис. 6.13 Настройки компиляции и загрузки

5. Нажмите **OK**.

6. Откройте ВП **crio thermostat (fpga)** и нажмите **Run**.

7. Нажмите **OK** в диалоговом окне **Beginning Compile**.

8. Нажмите **OK** в диалоговом окне **Successful Compile Report**

по окончании компиляции.

Теперь ВП запущен на целевом устройстве FPGA. Вводите другие значения температуры на лицевой панели VI. Если приложение включает вентилятор, загорается диод LED1 на модуле NI 9474.

При включении нагревателя загорается диод LED0 на модуле NI 9474.

Создание Host VI для CompactRIO

Мы будем использовать host VI для взаимодействия с FPGA VI и преобразования бинарных данных в температурные единицы. Host VI может быть запущен как на контроллере реального времени, так и на компьютере под управлением Windows. Выполните следующие действия для создания host VI.

1. Запустите LabVIEW.
2. Кликните на **crio thermostat.lvproj** в разделе **Open** окна **Getting Started**.
3. В окне **Project Explorer** кликните правой кнопкой мыши по названию базовой системы, например **My Computer** или **RT CompactRIO Target (0.0.0.0)**, и выберите **New»VI**.
4. Выберите **File»Save As** откроется окно **Name the VI**.
5. Введите **crio thermostat (host)** в поле **File name**.
6. Нажмите **OK**.
7. Поместите узел **Open FPGA VI Reference** на блок-диаграмму нового ВП. Мы будем использовать эту функцию для реализации связи с ВП, который загружается на целевое устройство FPGA. Функция **Open FPGA VI Reference** должна использоваться в каждом host VI.
8. Кликните левой кнопкой мыши по узлу **Open FPGA VI Reference** и выберите **Select VI (Выбрать ВП)**. Появится диалоговое окно **Select VI**.

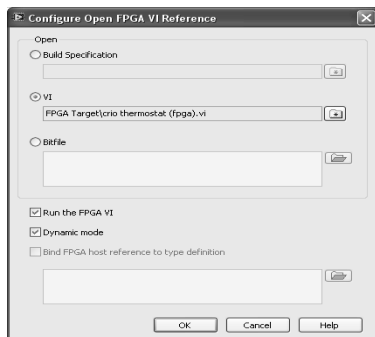


Рис. 6.14 Настройки Open FPGA VI

9. Найдите **crio thermostat (fpga) VI** и нажмите **OK**.



Рис. 6.15 Выбор FPGA

10. Поместите цикл **While Loop** на блок-диаграмму, справа от узла **Open FPGA VI Reference**.

11. Поместите узел **Close FPGA VI Reference** справа от цикла **While Loop**. Узел **Close FPGA VI Reference** должен использоваться в каждом **host VI**.

12. Поместите узел **Read/Write Control** внутрь цикла **While Loop**.

13. Соедините выход **FPGA VI Reference Out** узла **Open FPGA VI Reference** с входом **FPGA VI Reference In** узла **Read/Write Control**.

Считывание и преобразование текущего значения температуры

Выполните следующие действия для считывания фактического значения температуры и его преобразования в стандартные единицы.

1. Кликните правой кнопкой мыши на выход **Unselected** узла **Read/Write Control** и выберите **actual temperature**. Этот выход соответствует индикатору **actual temperature**, который вы создали в **FPGA VI**. Выход **actual temperature** возвращает данные бинарного типа. Эти данные нужно преобразовать в стандартные единицы температуры.

2. Выберите пункт **Select a VI** в палитре функций. Появится диалоговое окно **Choose the VI to open**.

3. Найдите директорию `LabVIEW 8.0\examples\ CompactRIO\Module Specific Examples\Analog Input\NI 9211\NI 9211 Support Files`

4. Выберите файл **Convert to Temperature (NI 9211).vi** и нажмите **OK**. Поместите ВП в цикл **While Loop**.

5. Соедините выход **actual temperature** узла **Read/Write Control** с входом **Thermocouple Channel** виртуального прибора **Convert to Temperature (NI 9211) VI**.

6. Кликните правой кнопкой мыши по узлу **Read/Write Control** и выберите **Add Control**.

7. Кликните правой кнопкой мыши по элементу **Unselected** и выберите **CJC**. Соедините выход **CJC** узла **Read/Write Control** с входом **CJC Channel** ВП **Convert to Temperature (NI 9211) VI**.

8. Добавьте выход **autozero** узла Read/Write Control, аналогичным образом. Соедините выход **autozero** функции Read/Write Control с входом **Autozero** ВП Convert to Temperature (NI 9211) VI.

9. Создайте константы для входов **Temperature Units** и **Thermocouple Type** ВП Convert to Temperature (NI 9211) VI.

10. Кликните правой кнопкой мыши по выходу **Linearized Temperature** ВП Convert to Temperature (NI 9211) VI и выберите **Create»Indicator**. Переименуйте индикатор в actual temperature.

Преобразование и запись требуемого диапазона температур

В модули вывода CompactRIO должны передаваться бинарные данные. Если Вы хотите вводить данные в стандартных температурных единицах, вы должны преобразовать их в бинарный вид до того, как host VI передаст значения установленного диапазона в FPGA VI.

1. Выберите **Select a VI** в палитре функций. Появится диалоговое окно **Choose the VI to open**.

2. Найдите директорию LabVIEW 8.0\examples\ CompactRIO\Module Specific Examples\Analog Input\NI 9211\NI 9211 Support Files.

3. Выберите файл **Convert to Binary (NI 9211).vi** и нажмите **ОК**. Поместите ВП в цикл While Loop.

4. Поместите еще один ВП Convert to Binary (NI 9211) VI в цикл While Loop.

5. Кликните правой кнопкой мыши по входу **Linearized Temperature** одного из

ВП Convert to Binary (NI 9211) VI и выберите **Create»Control**. Назовите элемент управления low target temperature.

6. Создайте элемент управления для входа **Linearized Temperature** для другого ВП Convert to Binary (NI 9211) VI и назовите его high target temperature.

7. Соедините выходы **CJC** и **autozero** узла Read/Write Control с входами **CJC Channel** и **Autozero Channel** ВП Convert to Binary (NI 9211) VI

8. Соедините константы **Thermocouple Type** и **Temperature Units** с соответствующими входами ВП Convert to Binary (NI 9211) VI.

9. Поместите другой узел Read/Write Control в цикл While

Loop справа от первого.

Соедините выход **FPGA VI Reference Out** первого узла Read/Write Control с входом **FPGA VI Reference In** второго узла Read/Write Control.

10. Кликните правой кнопкой мыши по элементу **Unselected** и выберите **low target temperature**. Соедините выход **Thermocouple Channel (Binary)** первого ВП Convert to Binary (NI 9211) VI с входом **low target temperature** узла Read/Write Control.

11. Кликните правой кнопкой мыши по новому узлу Read/Write Control и выберите **Add Control**. Кликните правой кнопкой мыши по выходу **Unselected** и выберите **high target temperature**. Соедините выход **Thermocouple Channel (Binary)** второго ВП Convert to Binary (NI 9211) VI с входом **high target temperature** узла Read/Write Control.

12. Соедините выход **FPGA VI Reference Out** второго узла Read/Write Control с входом **FPGA VI Reference In** узла Close FPGA VI Reference.

13. Соедините кластеры **error in** и **error out**.

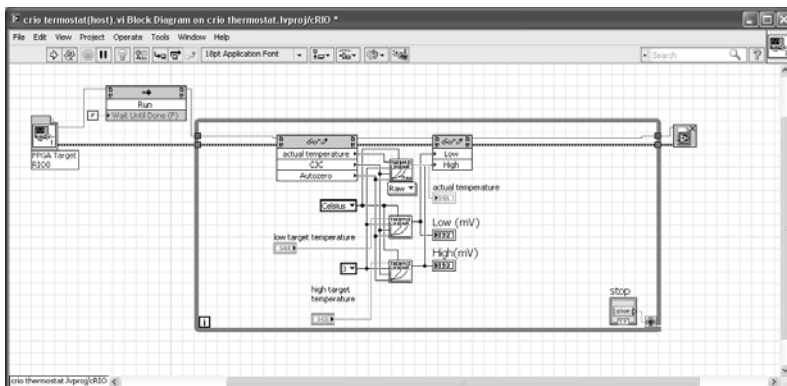


Рис. 6.16 Блок-схема управления температурой в режиме FPGA

Отчет по лабораторной работе должен содержать:

- 1) Описание хода выполнения.
- 2) Скриншот алгоритма программы и лицевой панели.
- 3) Выводы по работе различных режимов регулирования.

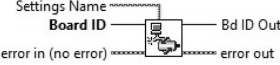
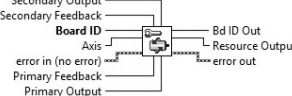
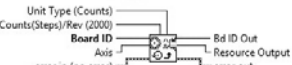
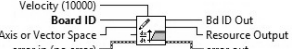
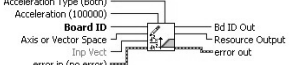
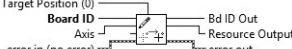
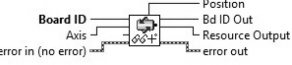
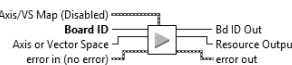
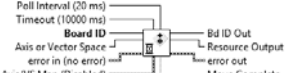
ЛАБОРАТОРНАЯ РАБОТА № 7. УПРАВЛЕНИЕ СЕРВОПРИВОДАМИ И ШАГОВЫМИ ДВИГАТЕЛЯМИ

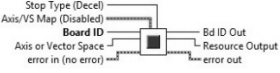
Сервопривод (следающий привод) — привод с управлением через отрицательную обратную связь, позволяющую точно управлять параметрами движения. Сервоприводом является любой тип механического привода (устройства, рабочего органа), имеющий в составе датчик (положения, скорости, усилия и т. п.) и блок управления приводом (электронную схему или механическую систему тяг), автоматически поддерживающий необходимые параметры на датчике (и, соответственно, на устройстве) согласно заданному внешнему значению (положению ручки управления или численному значению от других систем). Проще говоря, сервопривод является «автоматическим точным исполнителем» — получая на вход значение управляющего параметра (в режиме реального времени), он «своими силами» (основываясь на показаниях датчика) стремится создать и поддерживать это значение на выходе исполнительного элемента. К сервоприводам, как к категории приводов, относится множество различных регуляторов и усилителей с отрицательной обратной связью, например, гидро/электро/пневмо- усилители ручного привода управляющих элементов (в частности, рулевое управление и тормозная система на тракторах и автомобилях), однако термин «сервопривод» чаще всего (и в данной статье) используется для обозначения электрического привода с обратной связью по положению, применяемого в автоматических системах для привода управляющих элементов и рабочих органов.

Шаговый электродвигатель — это синхронный бесщёточный электродвигатель с несколькими обмотками, в котором ток, подаваемый в одну из обмоток статора, вызывает фиксацию ротора. Последовательная активация обмоток двигателя вызывает дискретные угловые перемещения (шаги) ротора. Шаговые электродвигатели применяются в приводах машин и механизмов, работающих в старто-стопном режиме, или в приводах непрерывного движения, где управляющее воздействие задаётся последовательностью электрических импульсов, например, в станках с ЧПУ. В отличие от сервоприводов, шаговые приводы позволяют получать точное позиционирование без использования обратной связи от датчиков углового положения.

Таблица 7.1

Описание основных блоков палитры NI-Motion 73xx

Внешний вид функции	Описание и назначение
<p>Initialize Controller</p> 	<p>Инициализация контроллера в соответствии с настройками, указанными в Measurement & Automation Explorer, где Board ID – идентификационный номер контроллера, в данном случае равен 1.</p>
<p>Configure Axis Resources</p> 	<p>Настройка оси и ее обратной связи, где Axis – выбор оси, а Primary Feedback/ Secondary Feedback – первичная/вторичная обратная связь. Программа может выполняться как с обратной связью, так и без нее.</p>
<p>Load Counts/Steps per Revolution</p> 	<p>Задание количества отсчетов на оборот - Counts (Steps)/Rev.</p>
<p>Load Velocity</p> 	<p>Задание скорости вращения - Velocity.</p>
<p>Load Acceleration/Deceleration</p> 	<p>Задание типа ускорения/замедления и его численного значения. Один блок может выполнять функцию, как отдельно ускорения или замедления, однако по умолчанию значение устанавливается для обоих типов.</p>
<p>Load Target Position</p> 	<p>Задание целевой позиции для выбранной оси - Target Position.</p>
<p>Read Position</p> 	<p>Чтение мгновенного значения положения указанной оси - Position. Для разомкнутых контуров возвращается количество генерируемых шагов. Для замкнутого контуров функция преобразует первичную позицию обратной связи от графов к шагам, а затем возвращает значение этого шага.</p>
<p>Start Motion</p> 	<p>Запуск движения выбранной оси.</p>
<p>Wait for Move Complete</p> 	<p>Проверка завершения движения за определенный временной интервал – Timeout. Тайм-аут (10000 мс) представляет собой количество времени, в миллисекундах, желаемое для окончания движение.</p>

Внешний вид функции	Описание и назначение
<p style="text-align: center;">Stop Motion</p> 	<p>Остановка движения одной оси, нескольких осей, или несколько векторных пространств. Существуют три типа остановки Stop Type: Decelerate stop – продолжение движения до полной остановки (отключение питания), Halt stop - остановка с фиксацией оси, Kill stop – резкая остановка сервопривода, используя весь его ресурс, направленный в противоположную сторону (резкий стоп).</p>

Задание на лабораторную работу

1. Подключить контроллер.
2. Сконфигурировать ось (порядковый номер оси уточнить у преподавателя).
3. Написать программу для управления заданной осью в следующих режимах:
 - a. Проехать 10000 тиков.
 - b. Переехать на позицию 10000 тиков.
 - c. Крутиться со скоростью 10000 тиков в секунду и так 10 секунд.
 - d. Повернуться на угол в 90, 180, 270 и 360 градусов.
4. Для каждого из пунктов записать результаты в табл.: фактические (экспериментальные) и расчетные (теоретические) значения начальной и конечная координаты, скорости, ускорения; установить уравнение движения, рассчитать абсолютную и относительную погрешности между расчетным и фактическим значениями.
5. Выполнить индивидуальное задание (получить у преподавателя).

Пример индивидуального задания: Написать программу, которая на 5 секунд запускает вращение 1-ой оси, затем включает и запускает вращение 2-ой и 3-ей осей, далее при нажатии на кнопку стоп отключает 2-ю и 3-ю оси, и затем подает команду на 4 ось – повернуться на угол 90 градусов.

Отчет по лабораторной работе должен содержать:

- 1) Описание хода выполнения.
- 2) Скриншот алгоритма программы и лицевой панели.
- 3) Выводы по работе различных режимов регулирования.
- 4) Блок схему и комментарии к индивидуальному заданию.

Рекомендация: Для формирования последовательного запуска серии моторчиков рекомендуется использовать структуру Flat Sequence Structure. Flat Sequence Structure (структура последовательности) состоит из одного или нескольких поддиаграмм или кадров, которые выполняются последовательно. Программы выполняются по порядку слева на право, вход одного кадра может зависеть от вывода другого кадра.

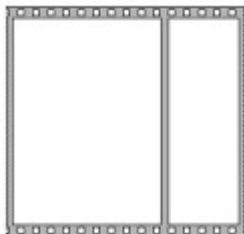


Рис. 7.1 Общий вид Flat Sequence Structure

РЕКОМЕНДУЕМЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Баран Е.Д.* Измерения в LabVIEW / Баран Е.Д., Морозов Ю.В. Новосиб.: НГТУ, 2010, 162 с.

2. *Корниенко В.Т.* Обеспечение безопасности передачи информации в радиотехнических системах с примерами в проектах LABVIEW: Учебное пособие. Таганрог : Южный федеральный университет, 2016, 80 с.

3. *Трэвис, Дж.* LabVIEW для всех: производственно-практическое издание / Дж. Трэвис, Дж. Кринг. – 3-е изд., перераб. и доп. М.: ДМК Пресс, 2008, 879 с.

СОДЕРЖАНИЕ

Лабораторная работа № 1. Выполнение вычислений с помощью циклов...	3
Лабораторная работа № 2. Работа со строковыми функциями	8
Лабораторная работа № 3. Измерение и анализ данных по каналу температура	10
Лабораторная работа № 4. Создание отчетных форм для MS WORD	13
Лабораторная работа № 5. Конфигурирование системы Compact FieldPoint	15
Лабораторная работа № 6. Конфигурирование ПЛК CompactRIO	23
Лабораторная работа № 7. Управление сервоприводами и шаговыми двигателями	38
Рекомендуемый библиографический список	42

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ УПРАВЛЕНИЯ И СБОРА ДАнных

*Методические указания к лабораторным работам
для студентов бакалавриата направления 15.03.04*

Сост.: *А.В. Бойков, Л.Н. Никитина, А.С. Симаков*

Печатается с оригинал-макета, подготовленного кафедрой
автоматизации технологических процессов и производств

Ответственный за выпуск *А.В. Бойков*

Лицензия ИД № 06517 от 09.01.2002

Подписано к печати 27.06.2019. Формат 60×84/16.
Усл. печ. л. 2,5. Усл.кр.-отт. 2,5. Уч.-изд.л. 2,2. Тираж 100 экз. Заказ 625. С 225.

Санкт-Петербургский горный университет
РИЦ Санкт-Петербургского горного университета
Адрес университета и РИЦ: 199106 Санкт-Петербург, 21-я линия, 2