

*МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ*

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Курганский государственный университет»

Кафедра «Экология и безопасность жизнедеятельности»

АВТОМАТИЗИРОВАННЫЕ КОМПЛЕКСЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

Методические указания

к комплексу лабораторных работ для студентов

направления подготовки

20.04.01 «Техносферная безопасность» направленность

(«Безопасность жизнедеятельности в техносфере»)

Курган 2019

Кафедра: «Экология и безопасность жизнедеятельности».

Дисциплина: «Автоматизированные комплексы обеспечения безопасности»
(направление 20.04.01 «Техносферная безопасность»)

Составил: канд. тех. наук, доц. Е. К. Карпов.

Утверждены на заседании кафедры

«26» июня 2019 г.

Рекомендованы методическим советом университета

«14» марта 2019 г.

ВВЕДЕНИЕ

Целью выполнения лабораторных работ по дисциплине «Автоматизированные комплексы обеспечения безопасности» является приобретение студентами знаний о современных методах создания автоматизированных комплексов обеспечения безопасности.

К задачам, выполняемым в рамках прохождения курса лабораторных занятий, относятся: ознакомление с областью применения компонентов автоматизации при разработке и создании устройств и комплексов, обеспечивающих защиту человека в техносфере. Изучение принципов работы основных элементов автоматизированных систем, особенностей их проектирования и разработки алгоритмов работы для обеспечения безопасности технических систем.

1 СИСТЕМА СИГНАЛИЗАЦИИ И СБОРА ИНФОРМАЦИИ С ДАТЧИКОВ БЕЗОПАСНОСТИ

Микроконтроллеры в комплексах обеспечения безопасности служат в качестве вычислительных центров, получая и накапливая информацию с датчиков и измерительных устройств а также осуществляя вывод информационных сигналов для пользователей системы и генерацию управляющих воздействий для исполнительных компонентов комплексов.

Задания:

1) соберите схемы с рисунков 1-4, запишите их программы на микроконтроллер и проверьте их работу;

2) используя материалы главы, соберите схему с LCD-монитором, кнопкой, потенциометром и фоторезистором. Запишите приведённую в главе для схемы программу и объясните её работу;

3) дополните схему одним из аналоговых датчиков, измеряющих величины факторов безопасности и допишите программу таким образом, чтобы при превышении уровня сигнала с него выводилось соответствующее сообщение на LCD-дисплей и загорался светодиод, подключенный к 13 ножке микроконтроллера.

Для того чтобы написать программу и записать её на микроконтроллер, необходимо выполнить следующую последовательность действий:

1) включить на компьютере исполнительный файл `arduino.exe` – это среда предназначена для разработки программного обеспечения и взаимодействия с контроллером посредством последовательного соединения;

2) выбрать в выпадающем меню “Файл”->“Новый” или нажать комбинацию клавиш `Ctrl+N` для создания нового проекта;

3) написать программный код, который предполагается исполнять на микроконтроллере, одного из заданий или разработанный самостоятельно;

4) проверить и скомпилировать программу, нажав на кнопку “Проверить” или используя комбинацию клавиш `Ctrl+R`. Во время первой проверки программа предложит сохранить файл программы. Если проверка прошла успешно, то можно переходить к следующему пункту действий, в противном

случае необходимо проверить синтаксис и правильность написания логики программы ещё раз;

5) прежде чем загружать программу в микроконтроллер, необходимо выбрать порт его подключения к компьютеру, тип платы и процессора, если это необходимо. Все эти операции производятся в подпунктах выпадающего меню “Инструменты” на верхней панели программы (в нашем случае это будет Arduino Uno без выбора типа процессора);

6) если все предыдущие действия были выполнены без ошибок, то можно произвести загрузку программы на контроллер, нажав кнопку “Загрузка” или используя комбинацию клавиш Ctrl+U.

Для установления связи между компьютером и микроконтроллером посредством последовательного соединения необходимо выполнить следующую последовательность действий:

1) включить на компьютере исполнительный файл `arduino.exe` – это среда предназначена для разработки программного обеспечения и взаимодействия с контроллером посредством последовательного соединения;

2) выбрать порт подключения микроконтроллера к компьютеру, тип платы и процессора, если это необходимо. Все эти операции производятся в подпунктах выпадающего меню “Инструменты” на верхней панели программы (в нашем случае это будет Arduino Uno без выбора типа процессора);

3) в выпадающем меню “Инструменты” выбрать пункт “Монитор порта” или использовать комбинацию клавиш Ctrl+Shift+M;

4) в нижнем поле появившегося окна будет выводиться вся информация, передаваемая от микроконтроллера, чтобы передать информацию на него, необходимо ввести её в верхнее поле окна и нажать клавишу Enter или кнопку “Отправить”. Скорость соединения, включение автоматической прокрутки и выбор символа конца строки можно настроить в процессе соединения.

В случае переключения платы необходимо закрыть окно “Монитора порта” и открыть его заново.

Электрическая схема с кнопкой и диодом

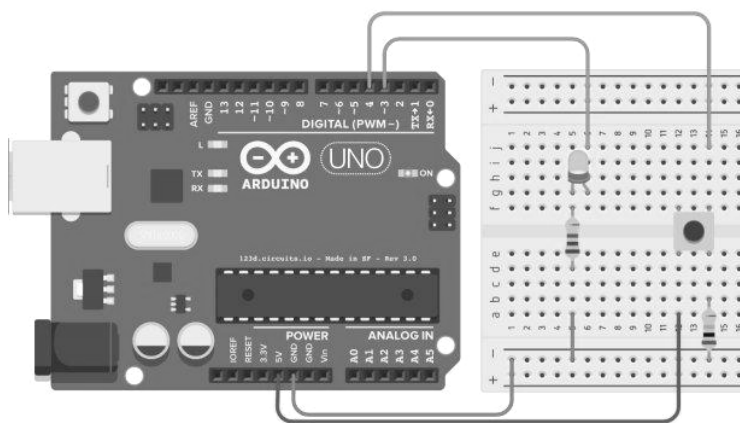


Рисунок 1 – Схема с одной кнопкой и одним диодом

Диодная индикация срабатывания кнопки. Текст программы:

```
void setup() {  
  pinMode(3, OUTPUT); // Третий контакт контроллера – выход (диод)  
  pinMode(4, INPUT); // Четвёртый контакт контроллера – вход (кнопка)  
}  
  
void loop() {  
  int a = digitalRead(4); // Чтение сигнала с кнопки в переменную  
  digitalWrite(3,a); // Подача значения переменной на диод  
}
```

Электрическая схема с фоторезистором и диодом

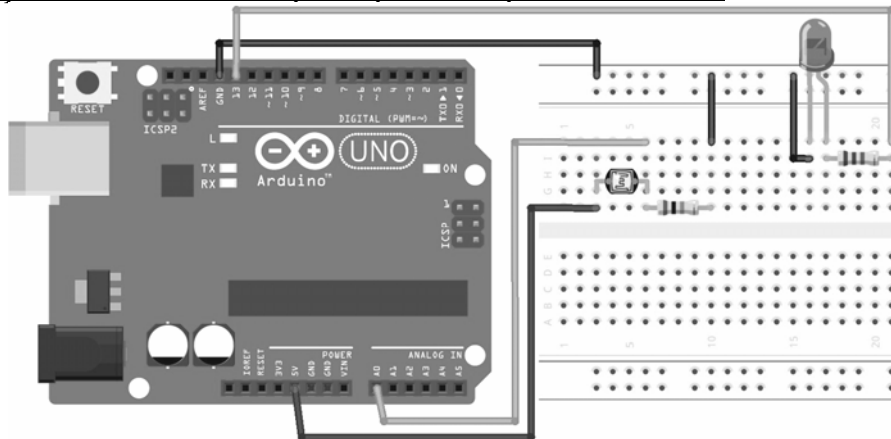


Рисунок 2 – Схема с фоторезистором и одним диодом

Включение диода в темноте и выключение на свету. Текст программы

```
void setup() {  
  pinMode(13, OUTPUT); // тринадцатый контакт – выход (диод)  
}  
  
void loop() {  
  //Если степень освещённости низкая (чтение аналогового порта A0 и  
  //сравнение с константой),  
  if (analogRead(A0) < 250)  
    digitalWrite(13, HIGH); //то включается светодиод  
  
  //Иначе  
  else  
    digitalWrite(13, LOW); //выключается светодиод  
}
```

Потенциометр (переменный резистор) и диод. Электрическая схема

Попробуйте заменить фоторезистор на потенциометр без изменения программы и посмотрите, как будет работать схема.

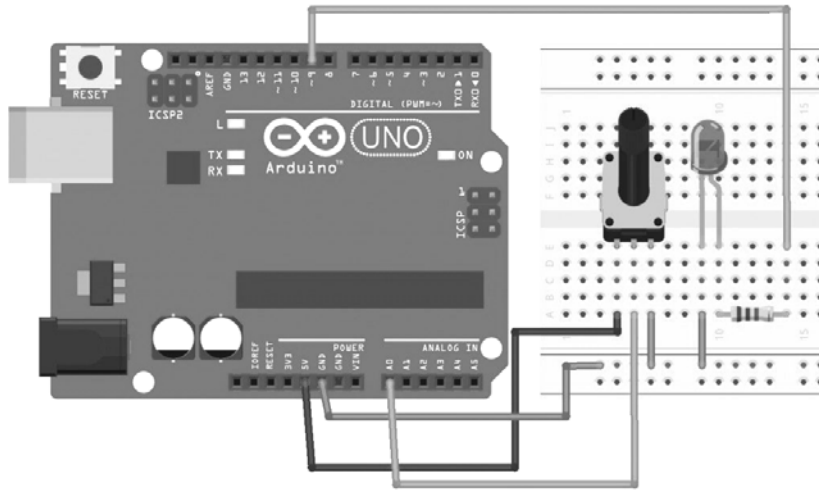


Рисунок 3 – Схема потенциометром и одним диодом

Управление двухстрочным шестнадцатисимвольным дисплеем LCD 1602

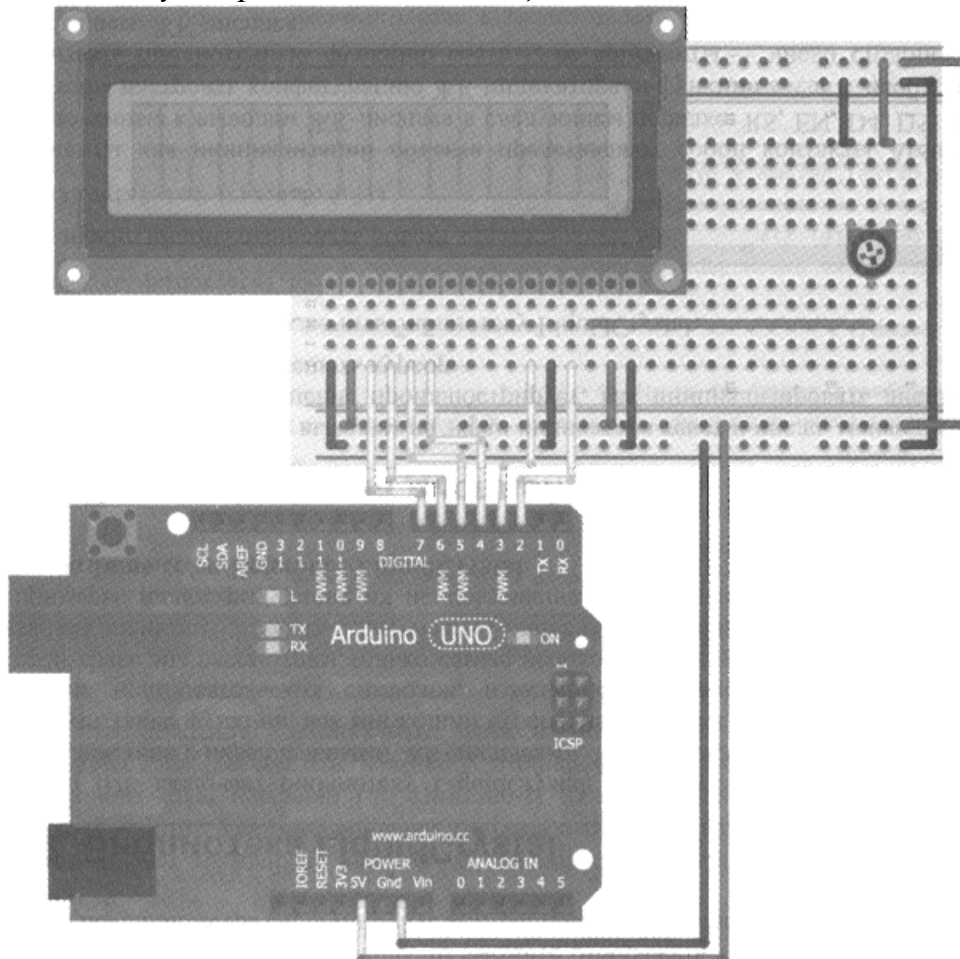


Рисунок 4 – Схема подключения дисплея с потенциометром уровня яркости
Текст программы секундного счётчика

```
#include <LiquidCrystal.h> //Подключение библиотеки мониторов
```

```
int time = 0; //Создание переменной счётчика, равного нулю
LiquidCrystal lcd(2,3,4,5,6,7); //Задание контактов подключения дисплея
```

```

void setup()
{
  lcd.begin(16,2); //Определение двухстрочного 16-символьного дисплея
  lcd.print("Display test"); //Вывод текста на дисплей
}

void loop()
{
  lcd.setCursor(0,1); //Установка курсора на первый символ второй строки
  lcd.print(time); //Вывод значения переменной
  delay(1000); //Задержка 1000 миллисекунд
  time++; //Инкрементирование переменной
}

```

Текст программы автоматизированной системы индикации и контроля безопасности по двум факторам (имитация при помощи датчика освещённости и потенциометра)

```

#include <LiquidCrystal.h> //Подключение библиотеки мониторов
LiquidCrystal lcd(2,3,4,5,6,7); //Задание контактов подключения дисплея
int display_num = 0;
int sys_halt = 0;

void setup() {
  lcd.begin(16,2); //Определение двухстрочного 16-символьного дисплея
  lcd.print("АКОВ lab №1"); //Вывод текста на дисплей
  pinMode(12, INPUT); // 12 контакт контроллера – вход (кнопка)
}

void loop() {
  if (sys_halt == 1)
  {
    lcd.setCursor(0,1);
    lcd.print("SYSTEM HALT!");
  }
  else
  {
    int fact_1 = analogRead(A1);
    int fact_2 = analogRead(A2);
    if (fact_1 > 500 || fact_2 > 500)
      sys_halt = 1;
    if (digitalRead(12) == HIGH)
      display_num ++;
    if (display_num > 1)
      display_num = 0;
  }
}

```

```

if (display_num == 0)
{
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print(fact_1);
}
if (display_num == 1)
{
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print(fact_2);
}
}
}

```

2 КОНТРОЛЬ ПОДДЕРЖАНИЯ ТЕМПЕРАТУРНОГО РЕЖИМА И ИНДИКАЦИЯ СОСТОЯНИЯ СИСТЕМЫ

В этой лабораторной работе рассматривается применение программируемых логических контроллеров в комплексах обеспечения безопасности на примере программно-технического комплекса «КОНТАР».

В состав системы управления входят контроллер МС8.3 и модуль расширения дискретных входов и выходов ME16 производства ОАО "Московский завод тепловой автоматики". Стенд оснащён четырьмя светодиодами, кнопкой и двумя переключателями, что позволяет дополнить лабораторные работы, выполняемые на нём, вариативной частью. Модули, из которых состоит стенд, могут быть применены в реальной автоматизированной системе управления. Для удобства они закреплены на планшете, произведена разводка всех соединений, цифровые выходы модулей для отображения их состояния нагружены на светодиоды, аналоговый выход соединён с негабаритным аналоговым механизмом. Аналоговый вход подсоединён к датчику температуры.

Контроллер МС8.3, входящий в состав комплекса «КОНТАР», имеет 4 дискретных и 8 аналоговых входов, а также 8 дискретных и 2 аналоговых выхода и предназначен для автоматизированного управления, контроля и мониторинга разнообразных технологических процессов:

- на объектах ЖКХ;
- в системах HVAC (отопление, вентиляция и кондиционирование) ресторанов, офисных зданий, спортивных сооружений, образовательных и медицинских центров;
- в различных установках для производства стройматериалов, пищевой промышленности и т.п.

Контроллер позволяет осуществлять сбор информации от любых источников (датчики температуры, давления, расхода, тепло-, водо-, электросчетчики и т.п.) и передавать ее на верхний уровень с использованием

различных каналов связи, в том числе сети Интернет. Контроллер может использоваться автономно или быть включенным в сеть приборов «КОНТАР» в составе распределенной системы управления.

Вопросы программирования и программного обеспечения подробно рассмотрены в методических указаниях «Изучение среды программирования «КОНГРАФ» для разработки программ на языке функциональных блоков» по дисциплине «Микроконтроллеры и микропроцессоры в системах управления».

Проектирование автоматизированной системы управления включает следующие этапы:

- 1) разработка алгоритма управления;
- 2) симуляция алгоритма управления;
- 3) трансляция алгоритма управления.

Алгоритм управления разрабатывается в «КОНГРАФ» на языке функциональных блоков (язык FBD, Functional Blocks Diagram – один из пяти технологических языков программирования контроллеров международного стандарта IEC-61131-3).

Любую часть создаваемого алгоритма управления или весь проект в целом можно предварительно (до трансляции и загрузки исполняемого кода в модули ПТК «КОНТАР») проверить программно, т.е. выполнить симуляцию нужной части проекта. При этом, как правило, оперативно находятся ошибки в работе алгоритма, которые необходимо или желательно переработать.

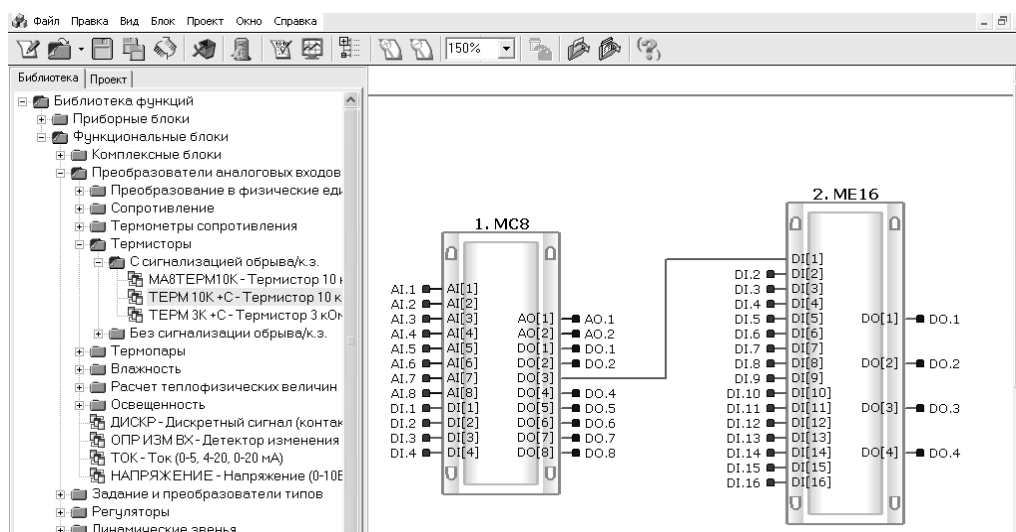


Рисунок 5 – Внешний вид программы «КОНГРАФ»

На рисунке 5 представлен внешний вид программы, в которой осуществляется процесс программирования на языке функциональных блоков. В левой части окна программы представлены в виде списка все доступные для программирования блоки, при помощи которых можно реализовать математические операции, дискретную логику, преобразование аналоговых и цифровых величин, а также различные законы автоматизированного управления. В правой части окна находится готовый проект. В этой программе осуществляется также компиляция и тестирование кода программы для контроллера.

В программе «КОНТАР-Консоль», внешний вид которой представлен на рисунке 6, производится загрузка программного кода в контроллер, а также осуществляется двухсторонняя передача параметров для их ручного контроля и отслеживания.

Интерфейс программы позволяет переводить выходы контроллера в ручной и автоматический режим. В первом случае алгоритм программы будет выполняться, но сигналы на выходы идти не будут.

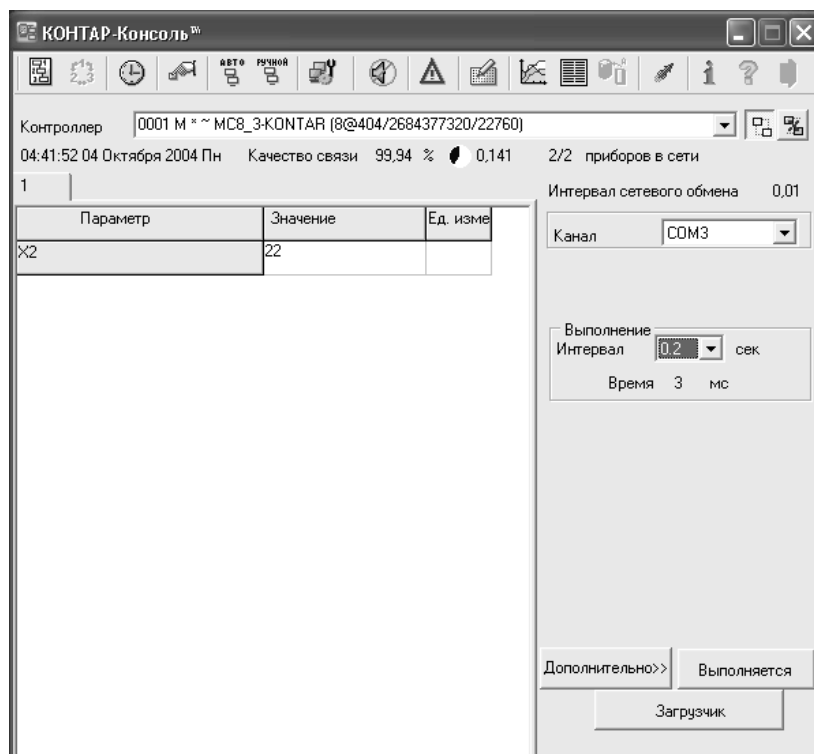


Рисунок 6 – Внешний вид программы «КОНТАР-Консоль»

Далее приведено описание процесса программирования контроллера для решения задачи поддержания температуры не ниже заданного значения посредством нагревательного элемента (лампочки). Измерение температуры будет осуществляться посредством термистора.

Задания: 1) повторить весь процесс программирования и выполнить программу для температуры 47 градусов Цельсия;

2) дополнить программу системой индикации: если температура ниже 47 градусов – горит красный левый диод, если выше 52 градусов – горит красный правый диод, если измеряемое значение температуры находится в диапазоне от 47 до 52 градусов и нагрев включён, то горит зелёный левый диод, если нагрев выключен, температура меньше 52 градусов и больше 47 градусов, то горят оба зелёных диода;

3) исследовать различные блоки библиотеки программы и реализовать исходный алгоритм управления, не используя блок «БОЛЬШЕ».

При создании нового проекта необходимо выбрать тип «Kontar» (рисунок 7) и указать его имя. Все проекты сохраняются в папке текущего пользователя компьютера в подпапках с именами проектов в качестве названий.

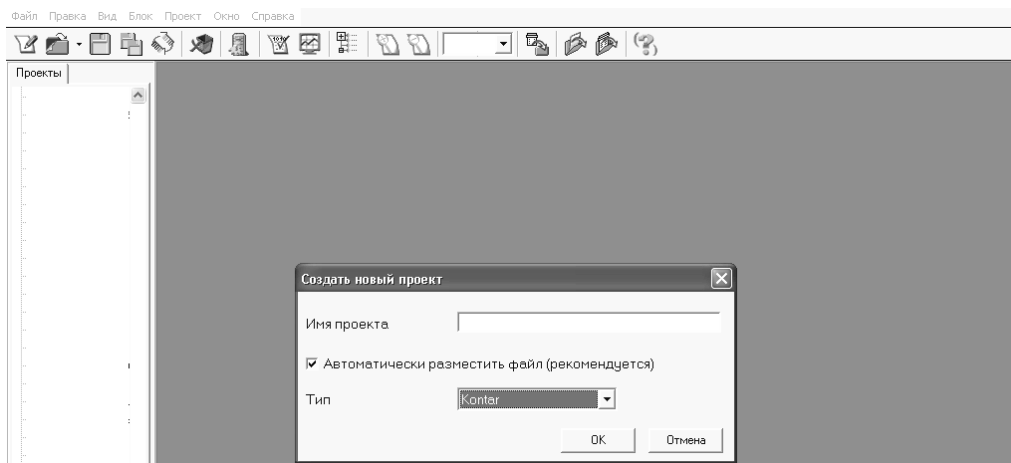


Рисунок 7 – Создание нового проекта

Первым делом в новый проект необходимо добавить наш контроллер из библиотеки элементов – подпапки «Контроллеры» (рисунок 8). В поле тип выбирается «Контроллер MC8.3», в противном случае программа не сможет на него загрузиться. Аналогичным образом добавляется релейный модуль ME16 (рисунок 9).

Теперь необходимо настроить режим MASTER для контроллера MC8.3. Для этого надо нажать на него правой кнопкой мыши, выбрать «Свойства...» и во вкладке «Блок» установить тип MASTER (рисунки 10, 11).

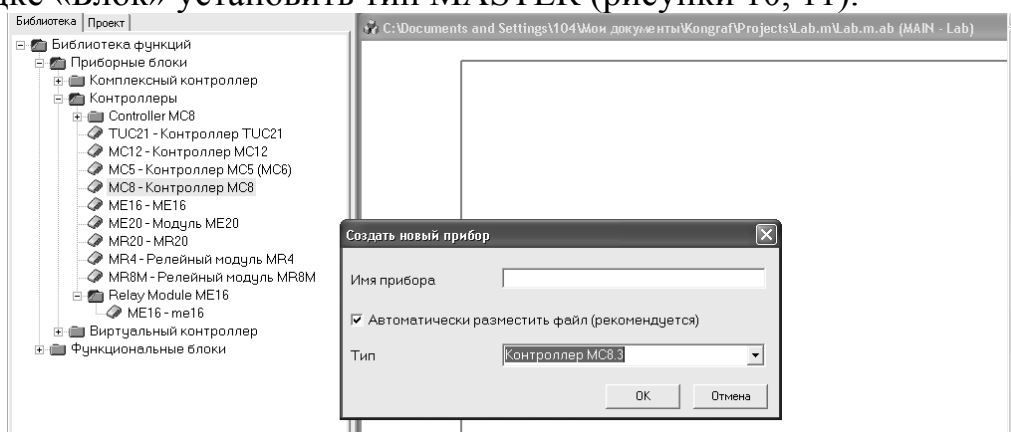


Рисунок 8 – Добавление микроконтроллера MC8.3

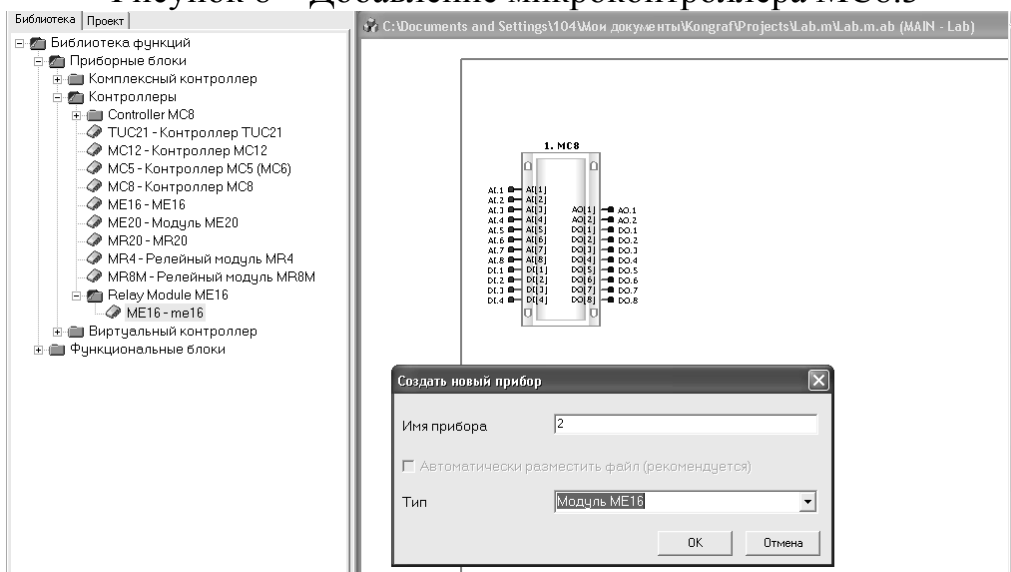


Рисунок 9 – Добавление релейного модуля ME16

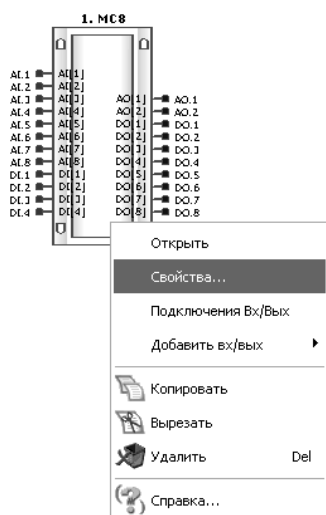


Рисунок 10 – Редактирование свойств контроллера

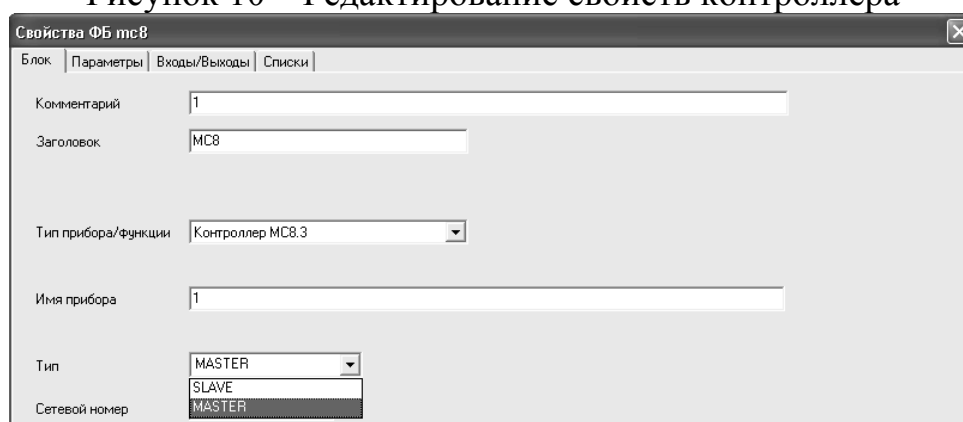


Рисунок 11 – Определение типа микроконтроллера (MC8.3 - MASTER, ME16 - SLAVE)

В настройках блока ME16 (рисунок 12) также необходимо проверить тип (SLAVE) и установить сетевой номер 2 (у MC8.3 – сетевой номер 1).

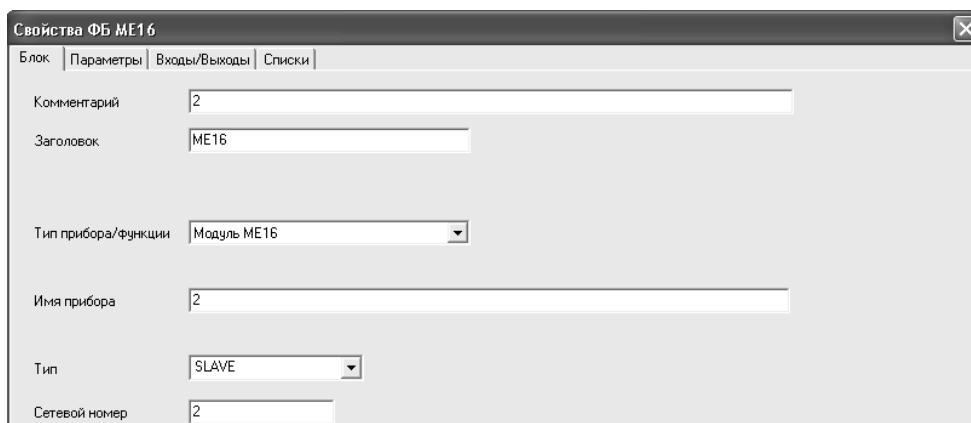


Рисунок 12 – Определение сетевого номера контроллера (MC8.3 - 1, ME16 - 2)

После этого щёлкаем левой клавишей мышки по блоку MC8.3 в правом рабочем поле программы и начинаем программировать логику работы системы управления. В библиотеке блоков из преобразователей аналоговых сигналов добавляем термистор, который используется в стенде ТЕРМ 10К с сигнализацией обрыва (рисунок 13). Физически термистор соединён с первым аналоговым входом контроллера, поэтому в программе также соединяем

первый вход AI[1] (Analog Input) с входом блока термистора. К выходу блока термистора YCEL (числовое значение измеряемой температуры) присоединяем логический блок «Больше» – он находится во вкладке математических функций

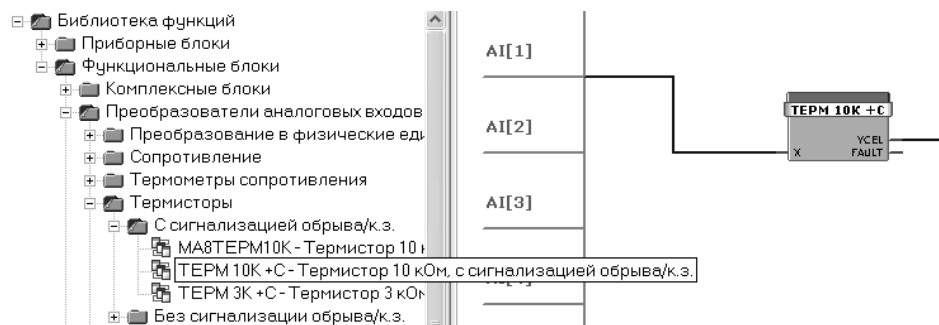


Рисунок 13 – Добавление термистора

библиотеки элементов (рисунок 14). Работает он следующим образом: когда на входе X1 значение больше, чем на входе X2, на выход блока Z подаётся логическая единица, в противном случае – ноль. Так как по условию задания необходимо поддерживать температуру на уровне 50 градусов, т.е. нагревательный элемент должен включаться, если измеренная температура ниже, то значение термистора надо подавать на второй вход логического блока.

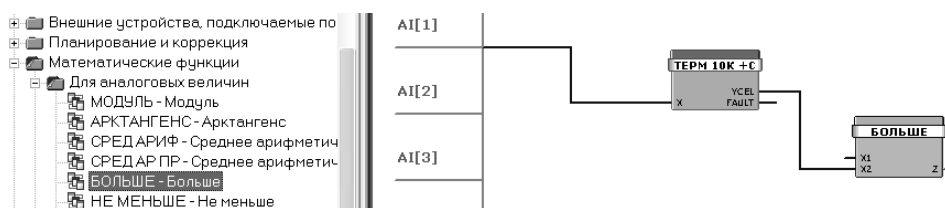


Рисунок 14 – Добавление логического блока «Больше»

Выход блока «Больше» подаётся на дискретный выход контроллера DO[1] (Discrete Output). К этому выходу физически подключён нагревательный элемент стенда – лампа накаливания (рисунок 15).

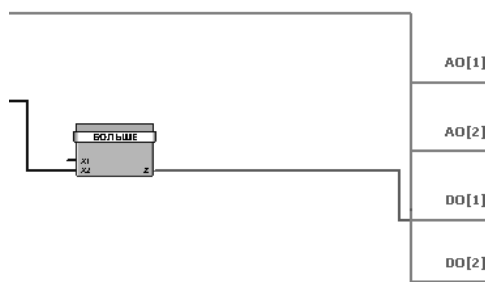


Рисунок 15 – Программное подключение выхода логического блока к дискретному выходу

Теперь осталось настроить логический блок «Больше», чтобы программа выполнялась так, как задумано. Для этого нажимаем на него правой кнопкой мыши, выбираем «Свойства...» и во вкладке «Параметры» устанавливаем галочку около «Константа» для Входа 1 – X1, затем вводим в поле «Значение» 50 (рисунок 16). Теперь по умолчанию значение температуры термистора, поступающее на вход 2, будет сравниваться с константой 50 – тем значением температуры, которое мы хотим поддерживать во время работы системы. Если значение X2 меньше 50, то на выходе блока Z будет логическая единица,

включающая нагревательный элемент. Если X2 больше 50, то на выходе будет ноль.

Для наглядности процесса работы изменим свойства блока термистора (рисунок 17). Во вкладке «Параметры» для YCEL создадим список, нажав на его пиктограмму, и введём в поле «Имя» слово «Температура». Теперь во время работы программы на терминал Консоли будет постоянно выводиться актуальное измеренное значение температуры. Это поможет проверить правильность работы логики программы и сделает интерфейс более ориентированным на пользователя.

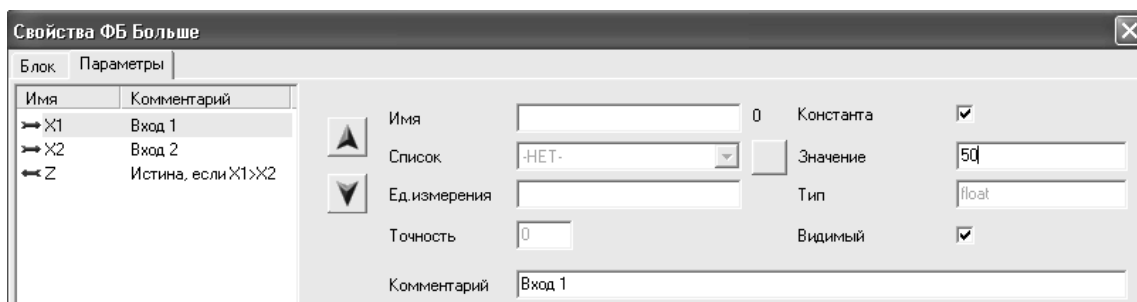


Рисунок 16 – Настройка логического блока «Больше»

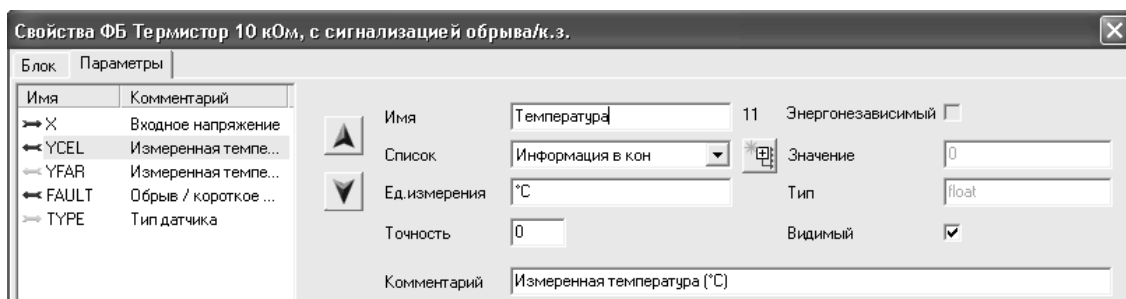


Рисунок 17 – Настройка параметров термистора

Чтобы записать проект в контроллер, сначала его надо скомпилировать: нажимаем на пиктограмму «Компилировать проект» и далее по инструкциям сохраняем его (рисунок 18). После этого запускаем программу «КОНТАР-Консоль» и подключаемся к контроллеру МС8.3 (рисунок 19). Выбираем в нижней части панели «Загрузчик» и скомпилированный файл программы, после этого жмём кнопку «Загрузить».



Рисунок 18 – Компиляция проекта

После окончания загрузки, если выходы контроллера установлены в автоматический режим, начнётся процесс поддержания заданного значения температуры при помощи двухпозиционного регулятора (рисунок 20).

Для добавления индикации при выполнении программы необходимо подавать на диоды стенда соответствующие сигналы – логические нули и единицы, в зависимости от логики программы.

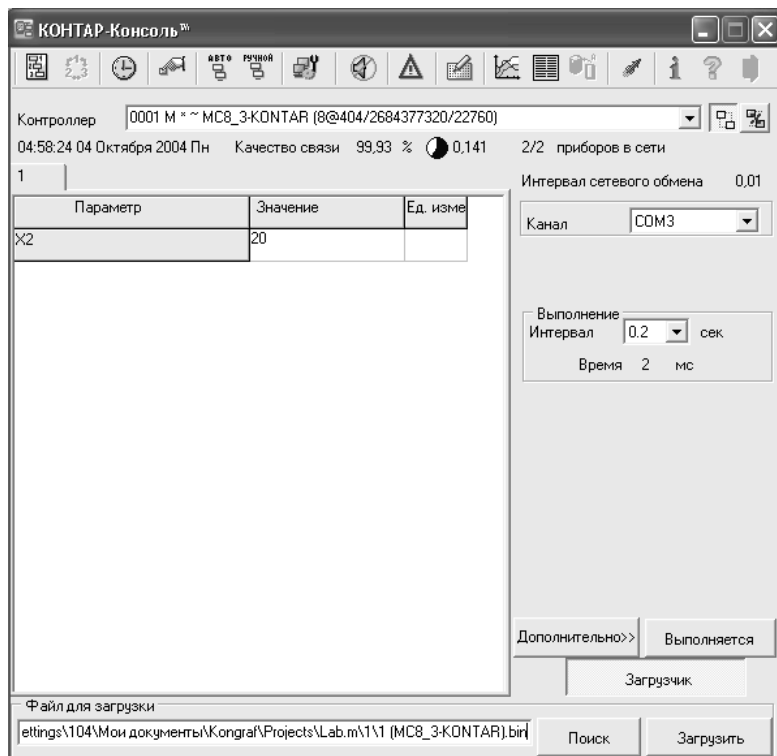


Рисунок 19 – Загрузка программы в контроллер

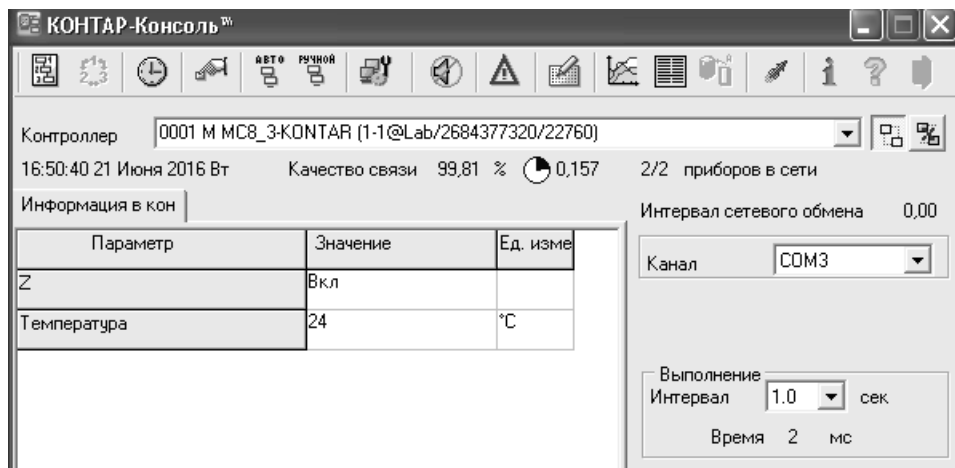


Рисунок 20 – Процесс выполнения программы

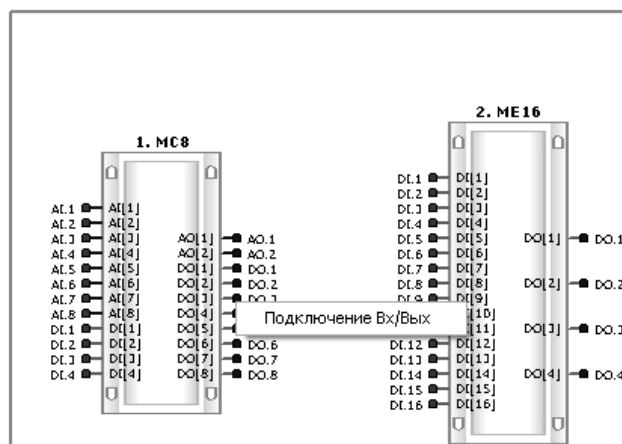


Рисунок 21 – Настройка связи между микроконтроллером и релейным модулем

Все диоды физически подключены к модулю дискретных сигналов ME16 – выходам DO[1-4]. Чтобы подавать на них сигналы, необходимо программно

соединить блоки МС8.3 и МЕ16. Для этого в поле программы нажимаем правой кнопкой на один из дискретных выводов МС8.3. Появится выпадающее поле «Подключение Вх/Вых» (рисунок 21), нажав на которое необходимо будет установить значение в режим «НЕТ» (рисунок 22). Можно использовать любой из дискретных выходов, но не стоит забывать, что к первому из них подключен нагревательный элемент, и отключать его таким образом не стоит.

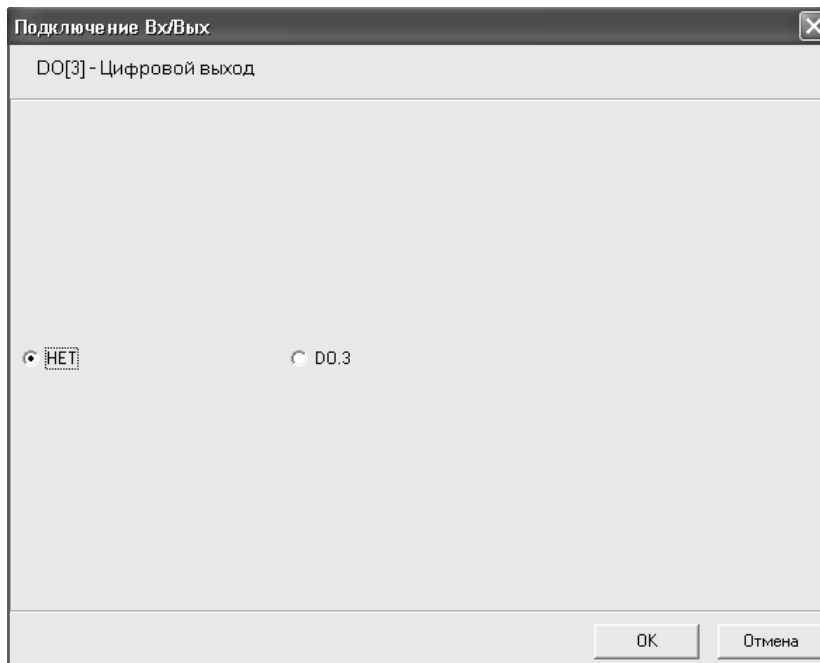


Рисунок 22 – Переключение режимов входов и выходов модулей контроллера

Аналогичным образом необходимо перенастроить один из входов модуля дискретных сигналов МЕ16. После этого их можно будет соединить таким же образом, как и обычные блоки программы и внутри блока МЕ16: подать данный сигнал прямо на один из дискретных выходов, к которым физически подключены диоды (рисунок 23).

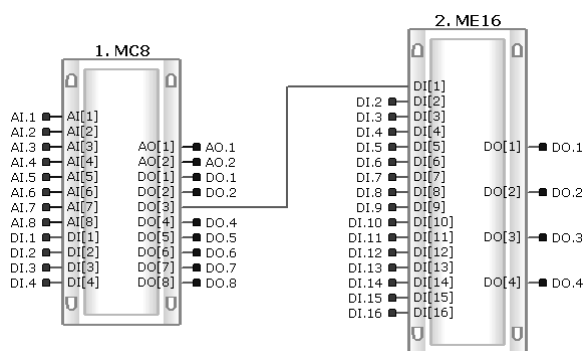


Рисунок 23 – Программное соединение модулей контроллера

3 ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ И ЗАДАНИЯ ПОВЫШЕННОЙ СЛОЖНОСТИ

1 Выполните все задания из обеих лабораторных работ данных методических указаний.

2 Аналогом кнопок, использованных в первой лабораторной работе являются контактные датчики с дискретным выходом. Приведите примеры их применения в автоматизированных комплексах обеспечения безопасности.

3 В первой лабораторной работе использовались датчики, измеряющие различные факторы безопасности. Приведите примеры их применения и применения комплекса безопасности аналогичной структуры в реальных условиях.

4 Во второй лабораторной работе напишите программу таким образом, чтобы включение нагревательного элемента осуществлялось с кнопки, расположенной на панели оператора. В том случае, если температура будет превышена, должны загораться оба красных светодиода, а нагрев должен отключаться до перезагрузки контроллера.

СОДЕРЖАНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ..... | 3 |
| 1 СИСТЕМА СИГНАЛИЗАЦИИ И СБОРА ИНФОРМАЦИИ С ДАТЧИКОВ БЕЗОПАСНОСТИ..... | 3 |
| 2 КОНТРОЛЬ ПОДДЕРЖАНИЯ ТЕМПЕРАТУРНОГО РЕЖИМА И ИНДИКАЦИЯ СОСТОЯНИЯ СИСТЕМЫ..... | 8 |
| 3 ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ И ЗАДАНИЯ ПОВЫШЕННОЙ СЛОЖНОСТИ..... | 17 |

Карпов Егор Константинович

АВТОМАТИЗИРОВАННЫЕ КОМПЛЕКСЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

Методические указания
к комплексу лабораторных работ для студентов
направления подготовки
20.04.01 «Техносферная безопасность» направленность
(«Безопасность жизнедеятельности в техносфере»)

Авторская редакция

| | | |
|-----------------------------|-------------------|----------------------------|
| Подписано в печать 08.10.19 | Формат 60x84 1/16 | Бумага 65 г/м ² |
| Печать цифровая | Усл. печ. л. 1,25 | Уч.-изд. л. 1,25 |
| Заказ 141 | Тираж 25 | Не для продажи |

БИЦ Курганского государственного университета.
640020, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.