МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Допущено к защите

Руководитель проекта

\_\_\_\_\_\_\_\_/Клюкин В.Л./

(подпись) (Ф.И.О)

«\_\_»\_\_\_\_\_\_\_\_\_\_2022г.

Разработка устройства автоматизированной проверки блоков релейной логики

Пояснительная записка курсового проекта по дисциплине

«Комплекс знаний бакалавра в области программного и аппаратного обеспечения вычислительной техники»

ТПЖА.09.03.01.487 ПЗ

Разработал студент группы ИВТ-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М./

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В.Л./

Консультант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мельцов В.Ю./

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_/

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись)

Киров 2022

РЕФЕРАТ

Птахова А.М.. Разработка устройства автоматизированной проверки блоков релейной логики. ТПЖА.09.03.01.487ПЗ: Курс. проект/ ВятГУ, каф. ЭВМ; рук. Клюкин.. – Киров, 2022. - ПЗ 30с, 3 табл., 8 рис., 3 прил.

РАЗРАБОТКА УСТРОЙСТВА, АВТОМАТИЗИРОВАННАЯ ПРОВЕРКА, БЛОК РЕЛЕЙНОЙ ЛОГИКИ, ARDUINO

Программно-аппаратная модель, разработанная в рамках данного курсового проекта – устройство автоматизированной проверки блоков релейной логики.

В ходе выполнения курсового проекта был выполнен анализ проблемной области, проектирование и разработка программного и аппаратного обеспечения.

Оглавление

[ВВЕДЕНИЕ 4](#_Toc119928798)

[1. АНАЛИЗ ПРОБЛЕМНОЙ ОБЛАСТИ 5](#_Toc119928799)

[1.1. Обзор текущей аппаратной модели проверки 5](#_Toc119928800)

[1.1.1. Интерфейс пользователя 5](#_Toc119928801)

[1.1.2. Осуществление проверки блока 5](#_Toc119928802)

[1.1.3. Выявление недостатков 5](#_Toc119928803)

[1.2. Обзор аналогов 5](#_Toc119928804)

[1.2.1. КЛУБ-У - Комплексно локомотивное устройство безопасности унифицированное. 5](#_Toc119928805)

[1.2.2. САК – система автоматического контроля 6](#_Toc119928806)

[1.2.3. АСУ – автоматическая система управления 7](#_Toc119928807)

[1.3. Актуальность разработки 8](#_Toc119928808)

[1.4. Техническое задание 9](#_Toc119928809)

[2. РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА 13](#_Toc119928810)

[2.1. Разработка структуры устройства 14](#_Toc119928811)

[2.2. Разработка диаграммы переходов состояний 14](#_Toc119928812)

[3. РАЗРАБОТКА АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ 15](#_Toc119928813)

[3.1. Алгоритм диагностирования 15](#_Toc119928814)

[3.2. Алгоритм проверки 16](#_Toc119928815)

[3.3. Алгоритм выбора режима 17](#_Toc119928816)

[4. ИЗГОТОВЛЕНИЕ ОБРАЗЦА И ПРОВЕДЕНИЕ ТЕСТОВ 18](#_Toc119928817)

[5. РАЗРАБОТКА И ОТЛАДКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 20](#_Toc119928818)

[ЗАКЛЮЧЕНИЕ 20](#_Toc119928819)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 21](#_Toc119928820)

[Приложение А 22](#_Toc119928821)

[Приложение Б 23](#_Toc119928822)

[Приложение В 24](#_Toc119928823)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | | | | | |
|  |  |  |  |  |
| Изм. | Лист | № докум | Подпись | Дата |
| Разраб | | Птахова А. М |  |  | Разработка устройства для автоматизированной проверки блоков релейной логики | Литера | | | Лист | Листов |
| Пров | | Мельцов В. Ю |  |  |  |  |  | 4 | 30 |
|  | |  |  |  |  | | | | |
|  | |  |  |  |
| Реценз. | |  |  |  |

# ВВЕДЕНИЕ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 4 |
| Изм. | Лист | № докум | Подпись | Дата |

В современном мире всё чаще возникает необходимость в автоматизации рутинных задач, выполняемых человеком. Одной из таких задач является задача проверки некоторого устройства на исправность. Для решения данной задачи возможно использование микроконтроллера, способного управлять электрическими устройствами по средствам выполнения программы.

Основной целью данного курсового проекта является разработка устройства для автоматизированной проверки блоков релейной логики.

# 1. АНАЛИЗ ПРОБЛЕМНОЙ ОБЛАСТИ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 5 |
| Изм. | Лист | № докум | Подпись | Дата |

На данном этапе работы необходимо провести обзор текущей аппаратной модели проверки, выявить её недостатки, сделать обзор аналогов, обосновать актуальность разработки и сформировать техническое задание

## 1.1. Обзор текущей аппаратной модели проверки

### 1.1.1. Интерфейс пользователя

Интерфейс пользователя представляет собой набор тумблеров для управления состояниями на входах проверяемого устройства, набор ламп и светодиодов для отображения состояний на выходах проверяемого устройства.

### 1.1.2. Осуществление проверки блока

Проверка блока осуществляется полностью оператором, который задает положение тумблеров и смотрит изменение состояний ламп или светодиодов в соответствии с алгоритмом, описанном на бумаге.

### 1.1.3. Выявление недостатков

Основным недостатком данной модели является выполнение проверки полностью «в ручном» режиме. Это сказывается на времени выполнения проверки и возможности ошибки в результате человеческого фактора.

## 1.2. Обзор аналогов

### 1.2.1. КЛУБ-У - Комплексно локомотивное устройство безопасности унифицированное.

Устройство предназначено для обеспечения безопасности движения, предупреждения аварийных ситуаций.

Для проверки работоспособности клуб-у пользователь должен ввести команду «К71». В результате выполнения команды в информационной строке высветится ряд из 11 цифр и букв «123456789АВ», которые обозначают наличие и исправность соответствующего логического модуля.

### 1.2.2. САК – система автоматического контроля

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 6 |
| Изм. | Лист | № докум | Подпись | Дата |

САК решает задачу автоматического получения информации о значении параметров или их отклонении от заданных значений и представлении ее в удобном виде оператору или для использования в системах автоматики. Как правило, осуществляет проверку только одного устройства, к которому прикреплен.

Пример САК для малого числа контролируемых параметров объекта приведен на рисунке 1.

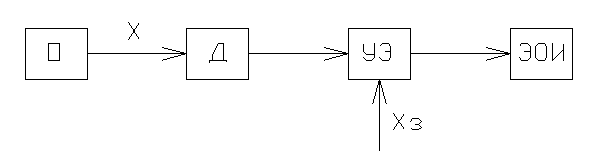


Рисунок 1 – САК для объекта с малым числом контролируемых параметров

В такой САК контролируемый параметр Х объекта О измеряется датчиком Д, сравнивается с заданным значение Х3 в управляющим элементе УЭ и результат измерения и контроля предоставляется оператору элементами отображения информации ЭОИ (световые и звуковые приборы).

### 1.2.3. АСУ – автоматическая система управления

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 7 |
| Изм. | Лист | № докум | Подпись | Дата |

Пример АСУ приведен на рисунке 2.



Рисунок 2 – АСУ

**На нижнем уровне система выполняет сбор, электрическую фильтрацию и АЦП сигналов с датчиков.**

**На верхнем уровне система выполняет следующие функции:**

1. Визуализация состояния технологического процесса;

2. Текущая регистрация характеристик технологического процесса;

3. Оперативный анализ состояния оборудования и **технологического процесса**;

Таким образом, нижний уровень реализует алгоритмы **управления** оборудованием, верхний – отображения и анализа полученных данных.

Сравнение аналогов приведено в таблице 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 8 |
| Изм. | Лист | № докум | Подпись | Дата |

|  |  |  |  |
| --- | --- | --- | --- |
|  | КЛУБ-У | САК | АСУ |
| 1.Выполняет проверку в автоматическом режиме | + | + | + |
| 2.Выполняет единственную операцию - проверку | - | + | - |
| 3.Есть возможность сравнения полученных данных с заданными | - | + | + |
| 4.Возможна проверка блоков релейной логики | - | - | - |

Таблица 1 – сравнение аналогов

## 1.3. Актуальность разработки

Существующее устройство проверки не имеет возможности работы в автоматическом режиме.

Существующие аналоги имеют ряд преимуществ перед существующим устройством: возможность проверки в автоматическом режиме, сравнения полученных данных с заданными, работы в одном режиме – режиме проверки. Но у них у всех отсутствует возможность проверки блоков релейной логики.

В результате было принято решение о разработке устройства для проверки блоков релейной логики, включающее в себя все преимущества аналогов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 9 |
| Изм. | Лист | № докум | Подпись | Дата |

## 1.4. Техническое задание

**1.4.1. Общее положение**

**1.4.1.1. Полное наименование системы**

Полное наименование системы: автоматизированная система проверки блоков релейной логики

**1.4.1.2. Определения, обозначения и сокращения**

Все сокращения и их расшифровка представлены в таблице 2

|  |  |  |
| --- | --- | --- |
| № | Сокращение | Расшифровка |
| 1 | Устройство | автоматизированная система проверки блоков релейной логики |
| 2 | Состояния | Значения 0 или 1, показывающие низкое (0 – 0.4 В) или высокое напряжение (2.4 – 5.0В) соответственно на ветвях схемы |
| 3 | Входы | Входные состояния, представляющие собой воздействие извне и влияющие на поведение блока |
| 4 | Выходы | Реакция блока на входные состояния |

Таблица 2 – Сокращения и их расшифровка

**1.4.2. Назначения и цели создания системы**

**1.4.2.1. Назначения системы**

Устройство предназначено для автоматизированной проверки блоков релейной логики.

Требования к устройству:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 10 |
| Изм. | Лист | № докум | Подпись | Дата |

- количество коммутируемых цепей не менее 16

- количество контролируемых цепей не менее 16

- количество цепей измерения постоянного напряжения 3

- максимальное напряжение по одной коммутируемой цепи не более 30В

- максимальный ток по одной коммутируемой цепи не более 2А

- максимальное напряжение по цепи измерения постоянного напряжения не более 30В

**1.4.2.2. Цели создания системы**

Основными целями устройства являются:

- повышение эффективности исполнения процесса проверки путем сокращения операций, выполняемых «вручную»

- повышение качества проверяемой продукции за счет уменьшения количества работ, выполняемых человеком

Для реализации поставленных целей система должна решать следующие задачи:

- распознавание входов и выходов

- установка состояний входов

- чтение состояний выходов

- проверка состояний входов и выходов в соответствии заданному алгоритму

- возможность выдачи результата

Пример алгоритма представлен в приложении А. Остальные алгоритмы проверки имеют такую же структуру.

**1.4.3. Характеристика объекта автоматизации**

Объектом автоматизации является пульт проверки блоков релейной логики.

Пульт проверки включает в себя:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 11 |
| Изм. | Лист | № докум | Подпись | Дата |

- тумблеры для установки состояний входов

- лампы и диоды для отображения состояний выходов

- инструкции для проверки

**1.4.4. Требования к системе**

**1.4.4.1. Требования к системе в целом**

**1.4.4.1.1. Требования к структуре и функционированию системы**

**1.4.4.1.1.1. Перечень подсистем, их назначения и основные характеристики**

В состав устройства должны входить следующие подсистемы:

- подсистема управления

- подсистема анализа

- подсистема операционного управления

Подсистема управления предназначена для подачи команд на блок управления, таких как установка состояний входов, чтения состояний выходов.

Подсистема анализа служит для проверки соответствия полученных данных с теми, которые должны были получиться, и распознавание полученных данных.

Подсистема операционного управления реализует установку состояний входов, чтение состояний выходов.

**1.4.4.1.1.2. Требования к способам и средствам связи для информационного обмена между компонентами системы**

Требования не предъявляются

**1.4.4.1.1.3. Требования к характеристикам взаимосвязей создаваемой системы со смежными системами**

Требования не предъявляются

**1.4.4.1.1.4. Требования к режимам функционирования системы**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 12 |
| Изм. | Лист | № докум | Подпись | Дата |

Требования не предоставляются

**1.4.4.1.1.5. Требования по диагностированию системы**

Диагностирование системы осуществляется установкой переключателя в соответствующий режим.

Режим диагностики включает в себя:

- диагностирование выходов при помощи задания их состояний посредством переключателей

- диагностирование входов при помощи включения светодиода при подаче на вход сигнала высокого напряжения

- диагностирование взаимодействия входов и выходов осуществляется посредством установки состояний выходов переключателями и последующей установки состояний соответствующих номерам выходов входов. Управляется через микроконтроллер.

**1.4.4.1.1.6. Перспективы развития, модернизации системы**

Устройство имеет возможность дальнейшего развития программного обеспечения. Также необходимо рассмотреть возможность масштабирования системы за счет увеличения количества коммутируемых цепей.

**1.4.4.1.2. Требования к численности и квалификации персонала системы**

Не предъявляются

**1.4.4.1.3. Показатели назначения**

Устройство должно обеспечивать возможность работы с не менее 16 коммутируемыми цепями, с не менее 50 контролируемыми цепями и с 3 цепями постоянного тока.

**1.4.4.2. Требования к функциям, выполняемым системой**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 13 |
| Изм. | Лист | № докум | Подпись | Дата |

Функция диагностирования выполняет следующее:

- чтение состояний выходов

- чтение состояний выходов и установку на номера входов, соответствующие номерам выходов, аналогичные состояния

Функция анализа выполняет следующее:

- установку состояние входа в состояние, заданное алгоритму

- чтение состояние выхода

- сравнения полученного значения с заданным

**1.4.4.3. Требования к видам обеспечения**

**1.4.4.3.1. Требования к информационному обеспечению системы**

Инструкции, жестко записанные в микроконтроллере, имеют вид конструкции switch- case. В ней в качестве параметра switch передается номер проверки, в операторах case идет выбор соответствующей проверки, выполнение функции анализа и переход к следующей проверки путем увеличение параметра switch на 1.

Данные поступают в виде сигналов логического нуля или единицы.

# 2. РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА

На данном этапе работы необходимо в соответствии с требованиями, сформулированными в техническом задании разработать структуру устройства и диаграмму переходов состояний

## 2.1. Разработка структуры устройства

Для обеспечения функционирования системы разработана обобщенная структура программно-аппаратного продукта, представляющая собой набор взаимосвязанных подсистем. Структурная схема устройства представлена на рисунке 3.

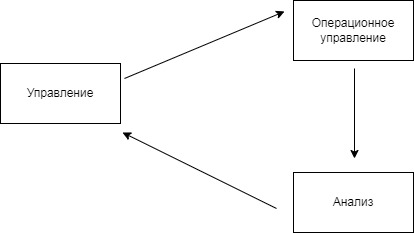


Рисунок 3 – Структурная схема устройства

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 14 |
| Изм. | Лист | № докум | Подпись | Дата |

## 2.2. Разработка диаграммы переходов состояний

Для описания поведения системы, демонстрации всех возможных состояний системы и изменением их в результате внешнего воздействия была разработана диаграмма переходов состояний, представленная на рисунке 4 и в приложении

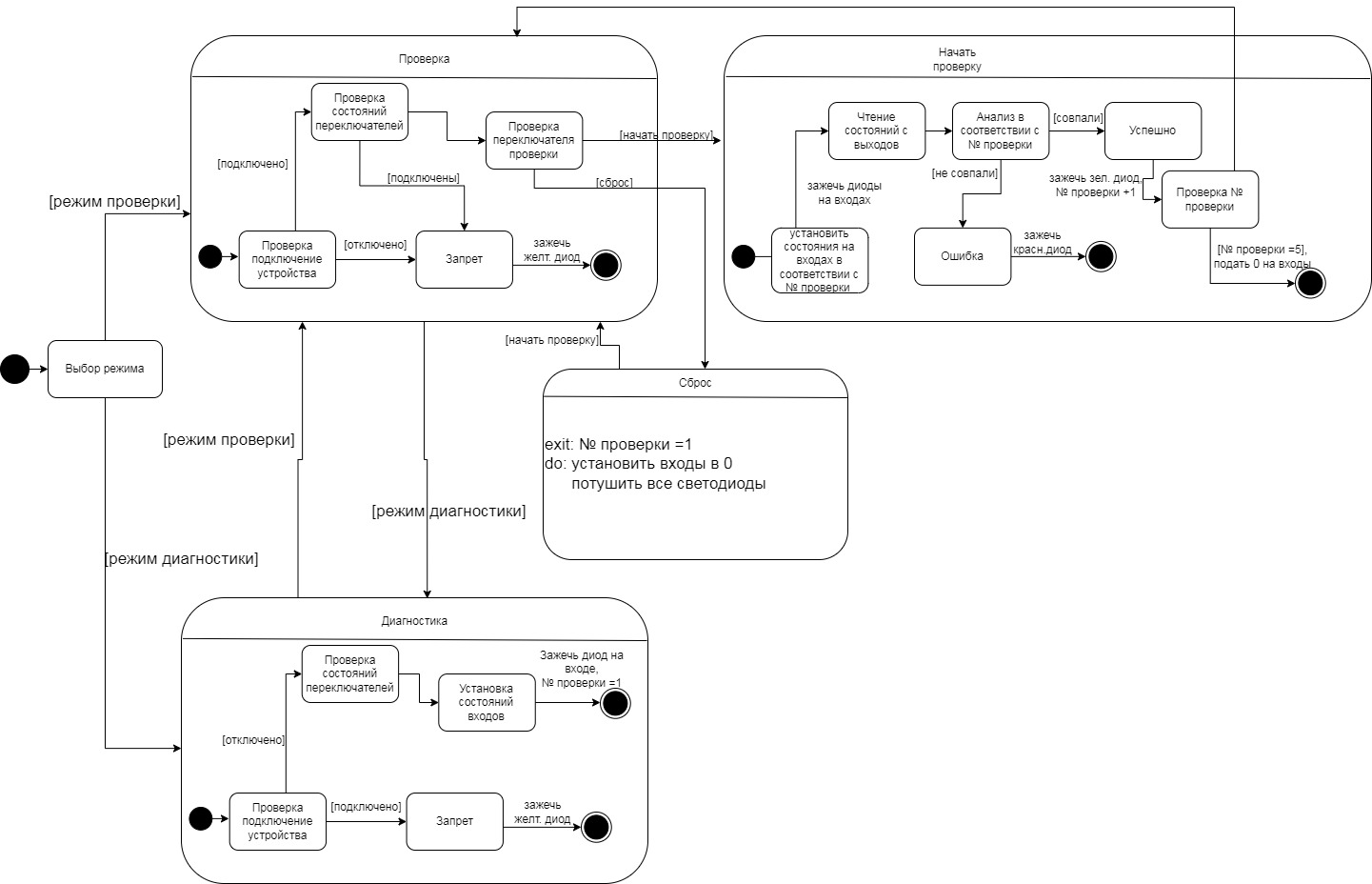


Рисунок 4 - Диаграмма переходов состояний

# 3. РАЗРАБОТКА АЛГОРИТМОВ ФУНКЦИОНИРОВАНИЯ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 15 |
| Изм. | Лист | № докум | Подпись | Дата |

Для реализации проверки блока необходимо разработать алгоритм проверки. Чтобы исключить возможность неисправности самого устройства проверки, необходим алгоритм диагностирования. Так как существует 2 различных режима работы устройства для проверки, необходимо реализовать алгоритм выбора соответствующего режима.

## 3.1. Алгоритм диагностирования

Схема алгоритма диагностирования представлена на рисунке 5.

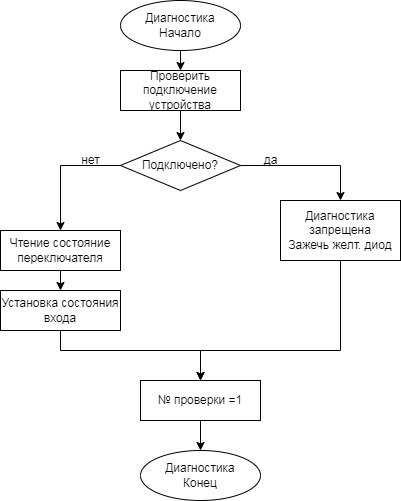


Рисунок 5 – Схема алгоритма диагностирования

## 3.2. Алгоритм проверки

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 16 |
| Изм. | Лист | № докум | Подпись | Дата |

Схема алгоритма диагностирования представлена на рисунке 6.

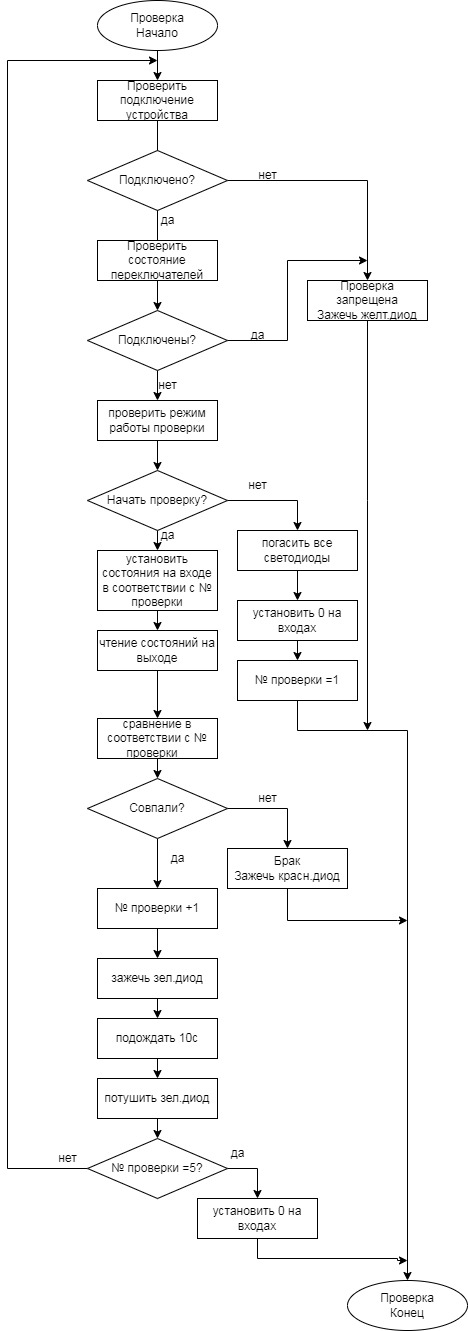


Рисунок 6 – Схема алгоритма проверки

## 3.3. Алгоритм выбора режима

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 17 |
| Изм. | Лист | № докум | Подпись | Дата |

Схема алгоритма выбора режима представлена на рисунке 7.

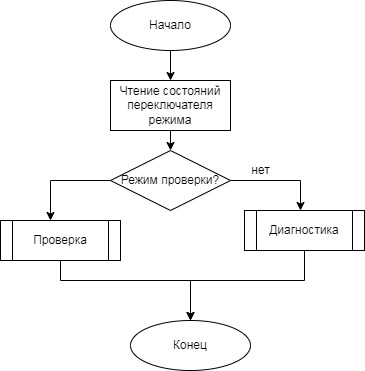


Рисунок 7 – Схема алгоритма выбора режима

# 4. ИЗГОТОВЛЕНИЕ ОБРАЗЦА И ПРОВЕДЕНИЕ ТЕСТОВ

Для реализации выбора режима диагностирования и возможностей проверки используется 2 разрядный dip переключатель. Первый разряд – выбор режима. Второй разряд – начать проверку или произвести сброс.

Для установки состояний выходов в режиме диагностирования используется 2 8-разрядных dip переключателей. Их подключение к плате осуществлялось при помощи 2 сдвиговых регистров 74hc165, так как они позволяют считывать данные, которые поступают на входы.

Для отображения состояний входов использовались 16 светодиодов, которые подключались к плате при помощи 2 сдвиговых регистров 74hc595.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 18 |
| Изм. | Лист | № докум | Подпись | Дата |

Для отображения состояний проверки используются 3 светодиода: зеленый, желтый и красный. Желтый загорается в тот момент, когда проверка или диагностирование запрещено. Зеленый – проверка прошла успешно. Красный – проверка выявила неисправности, устройство не исправно.

Схема разработанного устройства представлена в приложении Б и на рисунке 8.

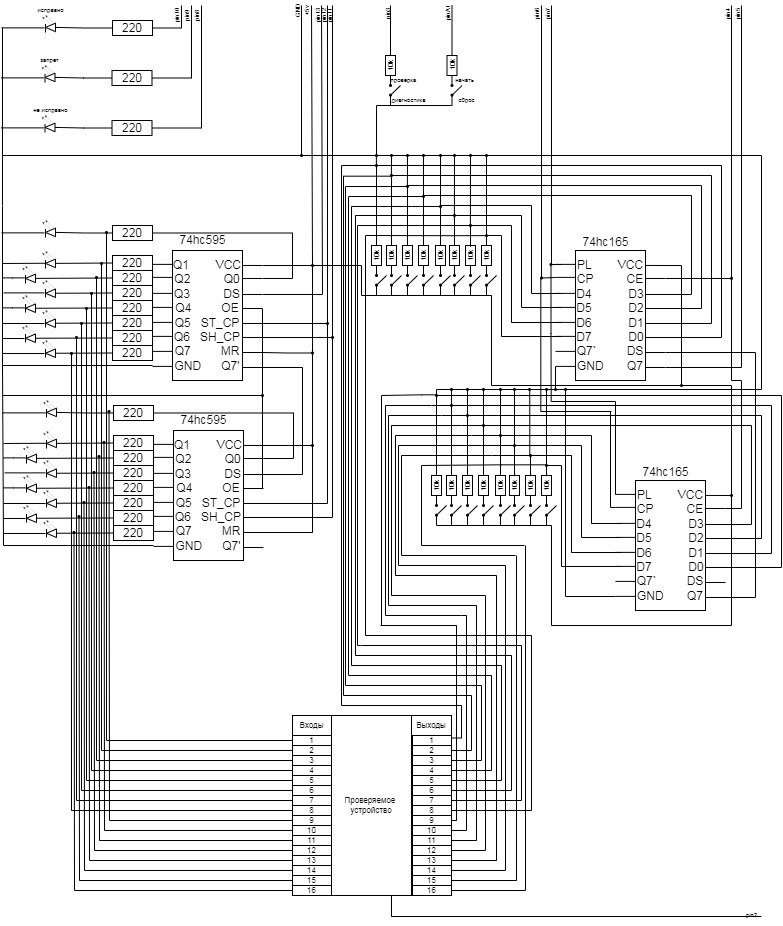


Рисунок 8 – схема устройства

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 19 |
| Изм. | Лист | № докум | Подпись | Дата |

# 5. РАЗРАБОТКА И ОТЛАДКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 20 |
| Изм. | Лист | № докум | Подпись | Дата |

Программировать микроконтроллеры можно на 2 языках программирования – Assembler, С. Их сравнение приводится в таблице 3.

|  |  |  |
| --- | --- | --- |
|  | Assembler | C |
| 1. ПО | - Notepad++  - GCC  - Tetminal и PuTTY  -  avrdude и openocd | - Arduino IDE |
| 2. Простота разработки | - | + |

Таблица 3 – сравнения языков программирования

Так как в техническом задании нет требований к скорости выполнения, а по остальным пунктам сравнения С имеет преимущества, этот язык и был выбран для написания программного обеспечения.

Листинг кода приведен в приложении В.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта было разработано устройство для автоматизированной проверки блоков релейной логики.

В качестве направления дальнейшего развития можно выбрать написание программы высшего уровня с возможностью задания алгоритма проверки, пошагового выполнения проверки и выдачей результата в виде отчета.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 21 |
| Изм. | Лист | № докум | Подпись | Дата |

1. Документация по arduino [Электронный курс] – режим доступа <https://docs.arduino.cc/>

# Приложение А

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 22 |
| Изм. | Лист | № докум | Подпись | Дата |

(обязательное)

**Пример алгоритма проверки**

На вход один подать высокое напряжение 🡪 на выходе 3 должно прийти высокое напряжение.

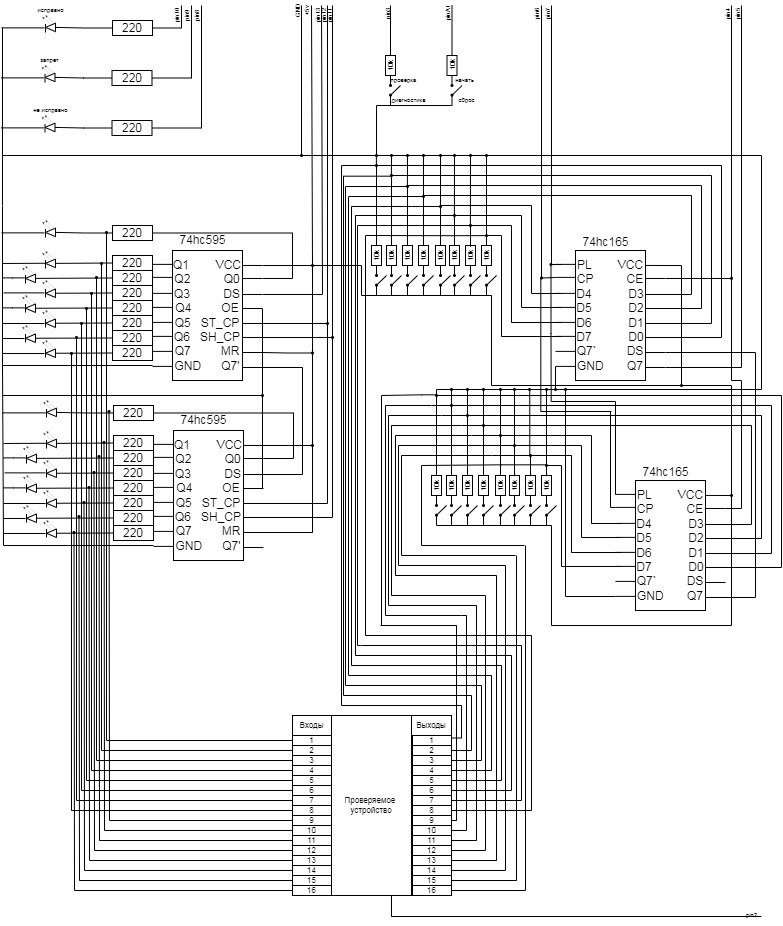
На вход один подать низкое напряжение 🡪 на выходе 3 должно прийти низкое напряжение.

# Приложение Б

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 23 |
| Изм. | Лист | № докум | Подпись | Дата |

(обязательное)

**Схема устройства**



# Приложение В

(обязательное)

**Листинг кода**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 24 |
| Изм. | Лист | № докум | Подпись | Дата |

//установка соответствия между названием и номером пина

//595 регистр

int latchPin=12;

int clockPin=11;

int dataPin=13;

//165 регистр

uint8\_t latchIn=7;

uint8\_t clockIn=6;

uint8\_t dataIn=5;

uint8\_t clockEnableIn=4;

//настройки

int regim =3;

int number=1;

int ledRed=8;

int ledYellow=9;

int ledGreen=10;

int vcc=2; //проверка питания

int for\_check=A1;

//инициализация

void setup() {

Serial.begin(9600); //подключение монитора порта

//установка пинов на вход или на выход

//led

pinMode(ledGreen, OUTPUT);

pinMode(ledRed, OUTPUT);

pinMode(ledYellow, OUTPUT);

//165

pinMode(latchIn, OUTPUT);

pinMode(clockIn, OUTPUT);

pinMode(dataIn, INPUT);

pinMode(clockEnableIn, OUTPUT);

//595

pinMode(latchPin, OUTPUT);

pinMode(dataPin, OUTPUT);

pinMode(clockPin, OUTPUT);

//настройки

pinMode(regim, INPUT);

pinMode(vcc, INPUT);

pinMode(for\_check, INPUT);

//начальная установка

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 25 |
| Изм. | Лист | № докум | Подпись | Дата |

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, 0);

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

}

//функция диагностики

void diagnostic()

{

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

//сдвинуть байт данных по одному биту за раз, записывая на вывод данных

shiftOut(dataPin, clockPin, LSBFIRST, 0);

//так как 16 разрядов - 2 байта, поэтому 2 функции

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

//устанавить защелку, чтобы захватить состояния выходов

digitalWrite(latchIn, LOW);

delayMicroseconds(5);

digitalWrite(latchIn, HIGH);

delayMicroseconds(5);

//считывание

digitalWrite(clockIn, HIGH); //без этого не считывает последний бит

digitalWrite(clockEnableIn, LOW); //установить запись, начать считывать бит

//сдвигаем байт данных по одному за раз, записывая на ввод данных

uint8\_t in1=shiftIn(dataIn, clockIn, LSBFIRST);

uint8\_t in2=shiftIn(dataIn, clockIn, LSBFIRST);

digitalWrite(clockEnableIn, HIGH); //сигнализировать, что больше не надо считывать

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, in2);

shiftOut(dataPin, clockPin, LSBFIRST, in1);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

delay(10);

}

//функция проверки устройства

void check(byte data, uint8\_t \*in1, uint8\_t \*in2)

{

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 26 |
| Изм. | Лист | № докум | Подпись | Дата |

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, data);

shiftOut(dataPin, clockPin, LSBFIRST, data);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

//устанавить защелку, чтобы захватить состояния выходов

digitalWrite(latchIn, LOW);

delayMicroseconds(5);

digitalWrite(latchIn, HIGH);

delayMicroseconds(5);

//считывание

digitalWrite(clockIn, HIGH); //без этого не считывает последний бит

digitalWrite(clockEnableIn, LOW); //устанавить запись, начать считывать бит

\*in1=shiftIn(dataIn, clockIn, LSBFIRST);

\*in2=shiftIn(dataIn, clockIn, LSBFIRST);

digitalWrite(clockEnableIn, HIGH); //сигнализировать, что больше не надо считывать

}

//функция проверки состояний переключателя в режиме проверки

void check\_begining(uint8\_t \*in1, uint8\_t \*in2)

{

//устанавливаем защелку, чтобы захватить состояния выходов

digitalWrite(latchIn, LOW);

delayMicroseconds(5);

digitalWrite(latchIn, HIGH);

delayMicroseconds(5);

//считывание

digitalWrite(clockIn, HIGH); //без этого не считывает последний бит

digitalWrite(clockEnableIn, LOW); //установить запись, начать считывать бит

\*in1=shiftIn(dataIn, clockIn, LSBFIRST);

\*in2=shiftIn(dataIn, clockIn, LSBFIRST);

digitalWrite(clockEnableIn, HIGH); //сигнализировать, что больше не надо считывать

}

//основная программа

void loop() {

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 27 |
| Изм. | Лист | № докум | Подпись | Дата |

uint8\_t in1;

uint8\_t in2;

int s1=digitalRead(regim); //считывание состояние переключателя режима

int s2=digitalRead(vcc); // считывание состояние питания

int s3=digitalRead(for\_check); //считываем состояние переключателя проверки

if (s1==0) //режим диагностики

{

if (s2==0) //не подключено проверяемое устройство

{

digitalWrite(ledRed, LOW);// все диоды потушить

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

diagnostic(); //диагностика

}

else //проверяемое устройство подключено

{

digitalWrite(ledRed, LOW);

digitalWrite(ledYellow, HIGH); //зажечь желтый диод - диагностика не возможна

digitalWrite(ledGreen, LOW);

}

number=1; //номер проверки =1

}

else //режим проверки

{

//подать нули на входы

digitalWrite(latchPin, LOW);//установить защелку, начать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, 0);

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

check\_begining(&in1, &in2); //проверить все ли переключатели, задающие выходы в режиме диагностики, выключены

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

delay(1000);

if ((in1!=0 ||in2!=0) && number==1 || s2==0 ) //переключатели не выключены в момент первой проверки или проверяемое устройство не подключено

{

digitalWrite(ledYellow, HIGH); //зажечь желтый диод - проверка не возможна

}

else //всё норм

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 28 |
| Изм. | Лист | № докум | Подпись | Дата |

{

if (s3==1)//начать проверку

{

switch(number) //номер проверки

{

case 1:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

check(0, &in1, &in2);//проверка

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==0 && in2==0)//сравнить полученные значения с эталонными

{

digitalWrite(ledGreen, HIGH); //проверка прошла

number++; //следующая проверка

} //if

else

{

digitalWrite(ledRed, HIGH);//не исправно

number=5;//заканчиваем проверку

}//else

delay(1000);

break;

}//1

case 2:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

delay(1000);

check(51, &in1, &in2);//проверка

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==51 && in2==51 )

{

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 29 |
| Изм. | Лист | № докум | Подпись | Дата |

digitalWrite(ledGreen, HIGH);

number++;

} //if

else

{

digitalWrite(ledRed, HIGH);

number=5;

}//else

delay(1000);

break;

}//2

case 3:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

delay(1000);

check(204, &in1, &in2);//проверка

//вывод в монитор порта

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==136 && in2==136 )

{

digitalWrite(ledGreen, HIGH);

number++;

} //if

else

{

digitalWrite(ledRed, HIGH);

number=5;

}//else

delay(1000);

break;

}//3

case 4:

{

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledGreen, LOW);

delay(1000);

check(255, &in1, &in2);//проверка

//вывод в монитор порта

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ТПЖА.09.03.01.487 ПЗ | Лист |
|  |  |  |  |  | 30 |
| Изм. | Лист | № докум | Подпись | Дата |

Serial.println(in1, DEC);

Serial.println(in2,DEC);

Serial.print('\n');

if (in1==221 && in2==221 )

{

digitalWrite(ledGreen, HIGH);

number++;

} //if

else

{

digitalWrite(ledRed, HIGH);

number=5;

}//else

delay(1000);

break;

}

default:

{

//записывание

digitalWrite(latchPin, LOW);//установить защелку, начинать передавать бит

shiftOut(dataPin, clockPin, LSBFIRST, 0); //сброс входов

shiftOut(dataPin, clockPin, LSBFIRST, 0);

digitalWrite(latchPin, HIGH); // сигнализировать, что больше не надо считывать

break;

}

}//switch

} // готовы к проверке

else //сброс

{

number=1;

digitalWrite(ledGreen, LOW);

digitalWrite(ledYellow, LOW);

digitalWrite(ledRed, LOW);

}

}//питание

} // regim = check

}