

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ОТЗЫВ
РУКОВОДИТЕЛЯ ВЫПУСКНОЙ РАБОТЫ МАГИСТРА

Выпускная работа магистра на тему
«Система автоматизированного управления газодинамическим стендом»

Выполнена студентом И.И. Ганиным
(инициалы, фамилия)

Факультет ИТЭ

Кафедра «НиМЭ». Группа 22ПДм1.

Направление подготовки, профиль .

11.04.04. «Электроника и наноэлектроника»,

«Нано- и микроэлектроника»

Квалификация (степень) Магистр

Руководитель Аношкин Юрий Владимирович
(фамилия, имя, отчество)

к.т.н., доцент кафедры «НиМЭ»
(должность и место работы)

Качества выпускника, выявленные в ходе выполнения выпускной работы
магистра

В ходе выполнения ВКР магистра Ганин И.И. показал навыки и умения к самостоятельной работе при проектировании автоматизированной системы управления газодинамическим стендом. Выпускник провел на высоком техническом уровне анализ литературных данных, проявил инициативу по внедрению автоматизированной системы для управления газодинамическим стендом для создания заданной концентрации контрольного компонента парогазовой смеси. При выполнении своих задач выпускник продемонстрировал навыки схемотехнического решения для достижения поставленной цели, а проведенное моделирование продемонстрировало работоспособность автоматизированной системы управления газодинамическим стендом.

Однако, в своем техническом решении Ганин И.И. использовал пять микропроцессоров, для управления одной газодинамической системой, что является дорогостоящим и не оптимизированным. Считаю, такое решение не является существенным недостатком, а требует оптимизации и упрощения схемы системы управления. С другой стороны автоматизированная система является работоспособной, что подтверждается моделированием.

Заключение

Выполненная выпускником ВКР магистра полностью соответствует техническому заданию. Степень соответствия подготовленности выпускника соответствует требованиям к результатам освоения программы магистратуры ФГОС ВО по направлению подготовки 11.04.04 «Электроника и наноэлектроника». На основании вышеизложенного считаю, что ВКР магистра заслуживает оценки «хорошо», а ее автор, Ганин И.И., присвоения степени «магистра» по направлению подготовки 11.04.04 «Электроника и наноэлектроника».

«17» июня 2024 г.

Руководитель

(подпись)

Ю.В. Аношкин

(инициалы, фамилия)

РЕЦЕНЗИЯ

на выпускную квалификационную работу

Ганина Ильи Игоревича

На тему: «Система автоматизированного управления газодинамическим стендом»,

по направлению 11.04.04 – Электроника и наноэлектроника

Целью рецензируемой работы является разработка автоматизированной системы управления газодинамическим стендом для создания парогазовых смесей с заданной концентрацией контрольного продукта, что позволяет отнести тему данного исследования к актуальным.

Применение газодинамических стендов для исследования чувствительности газовых сенсоров, а также контроль газодинамической среды в процессе выполнения поверки измерительных приборов является актуальной. Разработка новых технологий по созданию газовых сенсоров обеспечивают высокую чувствительность измерений к контрольному продукту парогазовых сред. Определение их чувствительности в широком диапазоне диктует новые требования к испытательному оборудованию.

Ключевой аспект практической значимости результатов работы заключается в возможности газодинамического стенда обеспечить управление концентрацией целевого компонента различных веществ в режиме реального времени с высокой точностью.

Рецензируемая работа содержит текстовый материал на 126 листов с 32 рисунками, библиография – 16 наименований. Работа выполнена в полном соответствии с техническим заданием, на высоком уровне, отражает уверенное владение автором положениями общетехнических и специальных дисциплин, современной техникой исследований.

Недостаточность обоснования использования технических средств для реализации автоматизированной системы управления газодинамическим стендом можно отнести к недостаткам работы, не снижающим научной и практической ценности работы.

В целом считаю, что выпускная квалификационная работа соответствует всем требованиям, предъявляемым к магистерским диссертациям по направлению подготовки 11.04.04 – «Электроника и наноэлектроника», заслуживает оценки «хорошо», а ее автор, Ганин Илья Игоревич, присвоения квалификации магистр по данному направлению обучения.

Начальник отдела АО «ПО «Электроприбор»



Комиссаров А.В.
18.06.24.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
информационных технологий
и электроники
(наименование)

Кафедра
«Нано- и микроэлектроника»
(наименование)

Направление подготовки

11.04.04. «Электроника и
наноэлектроника»
(код и наименование)

Магистерская программа

Нано- и микроэлектроника
(наименование)

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

Система автоматизированного управления газодинамическим стендом

Студент

22ПДм1
номер группы

 16.06.24
подпись, дата

Ганин Илья Игоревич
ФИО полностью

Научный руководитель

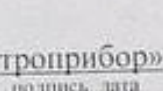
к.т.н., доцент, ПГУ
должность, наименование организации

 17.06.24
подпись, дата

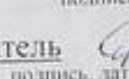
Ю.В. Аношкин
инициалы, фамилия

Рецензент

начальник отдела, АО «ПО «Электроприбор»
должность, наименование организации

 18.06.24 / А.В. Комиссаров
подпись, дата
инициалы, фамилия

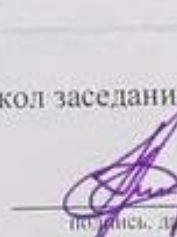
Нормоконтролер ст. преподаватель
должность

 17.06.24 Т.Н. Рыжова
подпись, дата
инициалы, фамилия

Работа допущена к защите в ГЭК (протокол заседания кафедры от 10.06.24 №10)

Заведующий кафедрой НнМЭ
профессор, ПГУ

должность, наименование организации

 10.06.24
подпись, дата

И.А. Пронин
инициалы, фамилия

Работа защищена с оценкой хор. (протокол ГЭК от №) 26.06.24 №11

Секретарь ГЭК

доцент, ПГУ
должность, наименование организации

 26.06.24 А.А. Вольсков
подпись, дата
инициалы, фамилия

2024
год выпуска

РЕФЕРАТ

Пояснительная записка содержит 127 листов, 1 таблицу, 32 рисунка, 5 приложений и 16 источников.

Ключевые слова: газодинамический стенд, парогазовая смесь, система, автоматизация, управления, температура, влажность, сброс, моделирование.

Актуальность работы связана с применением газодинамических стендов для исследования чувствительности газовых сенсоров, а также контроль газодинамической среды в процессе выполнения поверки измерительных приборов.

Цель работы – разработать автоматизированную систему управления газодинамическим стендом для создания парогазовых смесей с заданной концентрацией контрольного продукта.

Задачи:

- анализ компонентов электроники используемых в системах автоматизации;
- анализ применяемых измерительных приборов используемых в системе автоматизации;
- анализ исполнительных компонентов применяемых в качестве движущей силы в системах автоматизации;
- анализ микропроцессорной техники, которые упрощают систему управления технологическим процессом;
- разработать автоматизированную систему управления газодинамическим стендом.

Результат работы: автоматизированная система управления газодинамическим стендом, что позволяет упростить процесс создания парогазовых смесей с заданной концентрацией контрольного вещества.

СОДЕРЖАНИЕ

Введение.....	8
1 Обзор газодинамических систем и компонентов автоматизации.....	9
1.1 Газодинамический стенд.....	9
1.2 Шаговый двигатель компонент автоматизации	13
1.3 Типы температурных датчиков и принцип их работы	21
1.4 Газодинамические датчики потока	24
1.5 Типы датчиков влажности и принцип их работы.....	26
2 Разработка системы автоматизированного управления газодинамическим стендом.....	32
2.1 Микропроцессорная техника в системах управления технологическими процессами	32
2.2 Язык программирования.....	39
2.3 Микроконтроллеры серии 8051.....	40
2.4 Система автоматизированного управления газодинамическим стендом	47
2.5 Разработка модуля управления температурой.....	50
2.6 Разработка модуля управления сбросом парогазовой смеси.....	54
2.7 Разработка модуля управления разбавления воздухом	56
2.8 Разработка модуля управления влажностью воздуха	59
2.9 Разработка главного модуля управления	62
2.10 Автоматизированная работа газодинамического стенда	67

3 Техника безопасности при эксплуатации электрооборудования	68
Заключение	71
Список использованных источников	72
Приложение А.Задание на выпускную квалификационную работу магистра	74
Приложение Б.Схема электрическая функциональная, схема электрическая принципиальная	78
Приложение В.Листинг программ микроконтроллеров 5081	81
Приложение Г.Заявление и протокол проверки магистерской диссертации на оригинальность	115
Приложение Д.Презентация.....	119

ВВЕДЕНИЕ

Применение газодинамических стендов для исследования чувствительности газовых сенсоров, а также контроль газодинамической среды в процессе выполнения поверки измерительных приборов является актуальной. В настоящее время в связи с развитием новых технологий контроля парогазовых сред, а также создания новых газовых сенсоров контроля загрязнения окружающей среды является перспективным направлением в мире. Разработка новых технологий по созданию газовых сенсоров обеспечивают высокую чувствительность измерений к контрольному продукту парогазовых сред. Определение их чувствительности в широком диапазоне диктуют новые требования к испытательному оборудованию. Следовательно, постоянная модернизация и новые технические решения должны идти в ногу со временем.

Цель работы – разработать автоматизированную систему управления газодинамическим стендом для создания парогазовых смесей с заданной концентрацией контрольного продукта.

Для достижения цели необходимо выполнить следующие задачи:

- выполнить анализ компонентов электроники используемых в системах автоматизации, как для технологических процессов, так и для систем автоматики по безопасности;
- выполнить анализ применяемых измерительных приборов используемых в системе автоматизации;
- выполнить анализ исполнительных компонентов применяемых в качестве движущей силы в системах автоматизации;
- выполнить анализ микропроцессорной техники, которые упрощают систему управления технологическим процессом;
- разработать автоматизированную систему управления газодинамическим стендом.

1 Обзор газодинамических систем и компонентов автоматизации

1.1 Газодинамический стенд

Газодинамический стенд (ГДС) предназначен для измерения и анализа химических и физических свойств веществ, а также для подготовки лабораторных образцов парогазовых смесей (ПГС) путем разбавления и смешивания. Газодинамический стенд может использоваться в нормальных климатических условиях от 20 °С до 80 °С, в условиях моделирования воздействия ПГС на испытываемые вещества, а также для анализа эксплуатационных характеристик и показателей надежности технологии химического контроля. Полученные результаты используются для оценки концентрации ПГС.

Газодинамический стенд обеспечивает заданные функции с возможностью управления концентрацией целевого компонента различных веществ в режиме реального времени с заданной точностью. Газодинамический стенд состоит из диффузионного дозатора с изменяемой поверхностью испарения испытываемого вещества; смесителя газового потока для смешивания газа-носителя и чистого воздуха; капилляра и фильтр-поглотителя, включающего систему разбавления и несколько непрерывных модулей типа каскад.

Известна динамическая установка «Микрогаз-Ф» для приготовления газов и парогазовых смесей, содержащих заданные компоненты [1]. Установка предназначена для приготовления газов и парогазовых смесей с заданными массовыми концентрациями или объемными долями компонентов, используемых в промышленности, научных исследованиях и экологическом мониторинге. Установка «Микрогаз-Ф» представляет собой универсальный тип многоканального генератора сертифицированных газовых смесей, часто отказывающегося от использования парогазовых смесей (ПГС) в баллоне. Принцип действия установки основан на непрерывной генерации ПГС смесей термодиффузионным методом (с

использованием микропотоковых источников газа и пара) или методом динамического разбавления (смешивание газовых потоков при заданных объемных расходах) и их комбинации.

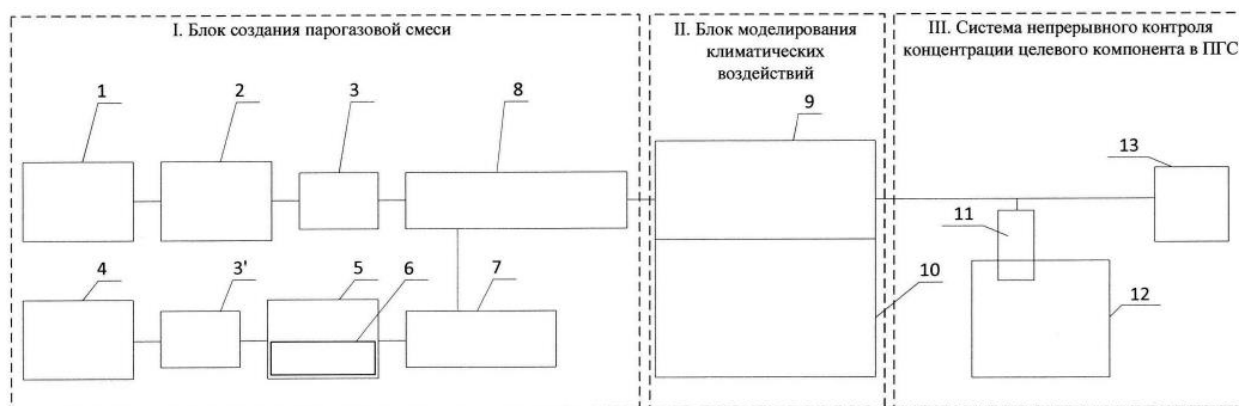
Известно газодинамическое устройство [2], предназначенное для создания ПГС стабильного состава высокотоксичных веществ для оценки качества технических средств систем мониторинга объектов по уничтожению химического оружия. Возможность изменять концентрацию целевых веществ в парогазовой смеси в диапазоне от 1×10^{-2} до 2×10^2 мг/м³ позволяет создавать смеси широкого спектра летучих веществ с возможностью изменения концентрации целевого вещества в парогазовой смеси в заданном диапазоне. Эта установка включает в себя дозатор в виде шприца Гамельтона и блок для создания парогазовых смесей заданной концентрации. Принцип работы газодинамической установки основан на использовании шприца целевого вещества, одновременно подключенного к регулятору расхода газа, и блока получения парогазовой смеси заданной концентрации, который представляет собой ряд испарителей, подключенных к динамическому блоку разбавителя.

Известен газодинамический испытательный стенд [3], включающий дозатор, газодинамическую установку для создания парогазовой смеси заданной концентрации и систему разбавления, состоящую из множества последовательно расположенных ступеней (каскадов) разбавления. Стенд позволяет готовить газовые смеси широкого спектра летучих веществ с воздухом в динамическом режиме.

Сущность метода смешения ПГС целевого компонента с воздухом заключается в испарении целевого компонента из жидкой фазы при заданной температуре в испарителе и тщательном смешении его с очищенным и осушенным потоком газа-носителя для получения первой парогазовой смеси. Для получения ПГС требуемой массовой концентрации последовательно подключается необходимое количество ступеней разбавления (каскадов), в которых исходная парогазовая смесь в строго дозированных количествах

смешивается с дополнительным потоком воздуха с заданной объемной долей.

Газодинамический стенд смешивает основное вещество в сухом воздухе в постоянной концентрации, и поддерживают это смешивание в течение длительного времени (рисунок 1.1).



1 – подложка блока очистки и осушки воздуха; 2 – компрессором; 3, 3' – регулятор расхода газа; 4 – системой подачи газа; 5 – диффузионный дозатор; 6 – термостат; 7 – каскад разбавления; 8 – смеситель газовых потоков; 9 – климатическая камера; 10 – испытываемый образец; 11 – отбор проб; 12 – хромато-масс-спектрометр; 13 – фильтр-поглотитель

Рисунок 1.1 – Структура газодинамического стенда

Газодинамический стенд работает следующим образом: атмосферный воздух из блока очистки и осушки воздуха 1 с компрессором 2 подается по газопроводу в смеситель газового потока 8 через регулятор газового потока 3, при этом точно регулируется необходимый объем воздуха и смесь паров и газов разбавление.

Газ-носитель (азот) подается из системы газоснабжения 4 по газопроводу к регулятору расхода газа 3' и подаваемый газ точно дозируется в диффузионный дозатор 5, куда вводится необходимое количество целевого вещества. Диффузионный дозатор 5 помещен в термостат 6 для поддержания оптимальной температуры испарения целевого вещества. Пары целевого вещества из внутреннего объема диффузионного дозатора 5 выдуваются газом-носителем, поступающим из регулятора расхода газа 3', и

направляются в каскад разбавления 7. Далее попадает в фильтр-поглотитель и после фильтрации выбрасывается в атмосферу, а остальной поток парогазовой смеси, прошедший через муфту, попадает в следующий каскад, где разбавление повторяется. Диаметр капилляра определяется расчетным путем, и количество каскадов разбавления может быть увеличено. Парогазовая смесь из каскадного разбавителя 7 по газопроводу поступает в смеситель газового потока 8, где осушенный и очищенный воздух смешивается с парогазовой смесью, разбавляя ее до необходимой концентрации. Смеситель газового потока 8 соединен винтовым соединением с теплообменным радиатором, расположенным в климатической камере 9. Парогазовая смесь, проходящая через теплообменный радиатор, на выходе разделяется на два потока: первый поток проходит через штуцер и поступает в испытуемый образец 10 в климатической камере 9, а второй поток парогазовой смеси проходит через газовую магистраль и непрерывно чередуется пробоотборником блок 11, через который поток парогазовой смеси поочередно направляется в адсорбционную трубку в термодесорбере 12. Далее адсорбционная трубка подключается к газовому хроматографу с масс-селективным детектором 12 и после десорбции целевой компонент направляется в блок газового хроматографа. Оставшееся количество парогазовой смеси, не участвующее в определении целевого вещества, по газовой магистрали поступает в фильтр-адсорберный комплекс 13 и после фильтрации выбрасывается в атмосферу. Система непрерывного попеременного отбора проб совмещена с газовым хроматографом-масс-спектрометром со встроенным термодесорбером для попеременного отбора проб ПГС на адсорбционную трубку и последующей десорбции на кварцевую капиллярную колонку газового хроматографа, что обеспечивает получение парогазовой смеси.

Количественное содержание целевого компонента в ПГС можно контролировать в режиме реального времени, определяя целевой компонент с помощью масс-спектрометра и определяя по результатам количественное

содержание целевого компонента в ПГС. Время анализа не превышает 5 минут. В момент анализа автоматически отбирается вторая адсорбционная трубка с сорбентом, и десорбция целевого компонента из второй трубки начинается сразу после завершения анализа первой пробы. Во время испытания циклический отбор проб из первой и второй трубок с сорбентом осуществляется непрерывно.

1.2 Шаговый двигатель компонент автоматизации

Шаговые двигатели – это электрические машины, предназначенные для преобразования электрической энергии электросети в механическую. Конструктивно они состоят из обмотки статора и магнитомягкого или жесткого ротора. Шаговые двигатели характеризуются прерывистым вращением, когда фиксированное число импульсов соответствует фиксированному числу шагов. Чаще всего они используются в станках с числовым программным управлением (ЧПУ), робототехнике, устройствах хранения и считывания данных [4].

В отличие от других типов машин, шаговые двигатели вращаются не непрерывно, а пошагово. Каждый шаг – это только часть общего вращения. Количество шагов, необходимых для полного вращения вала, зависит от способа подключения, марки двигателя и метода управления.

Преимущества использования шаговых двигателей заключаются в следующем:

- при использовании шаговых двигателей угол поворота соответствует количеству подач электрического сигнала, а полный крутящий момент и блокировка сохраняются даже при остановке вращения;
- точное позиционирование от 3 до 5 % от заданного шага;
- быстрый запуск, реверс и остановка;
- в отличие от коллекторных двигателей, они характеризуются высокой надежностью, так как в них отсутствуют элементы трения, по которым протекает ток;

- шаговые двигатели не требуют обратной связи для позиционирования;
- низкие скорости могут быть достигнуты для прямых нагрузок без редуктора;
- относительно низкая стоимость по сравнению с аналогичными сервоприводами;
- скорость вращения вала можно регулировать в широком диапазоне, изменяя частоту электрических импульсов.

К недостаткам использования шаговых двигателей относятся:

- возможны эффекты резонанса и проскальзывание шагового узла;
- отсутствие обратной связи, что может привести к потере контроля;
- потребляемая мощность не зависит от наличия или отсутствия нагрузки;
- сложность управления из-за особенностей схемы.

На рисунке 1.2 показаны четыре обмотки статора двигателя, которые расположены под углом 90° друг к другу. Таким образом, машина характеризуется шагом в 90° .

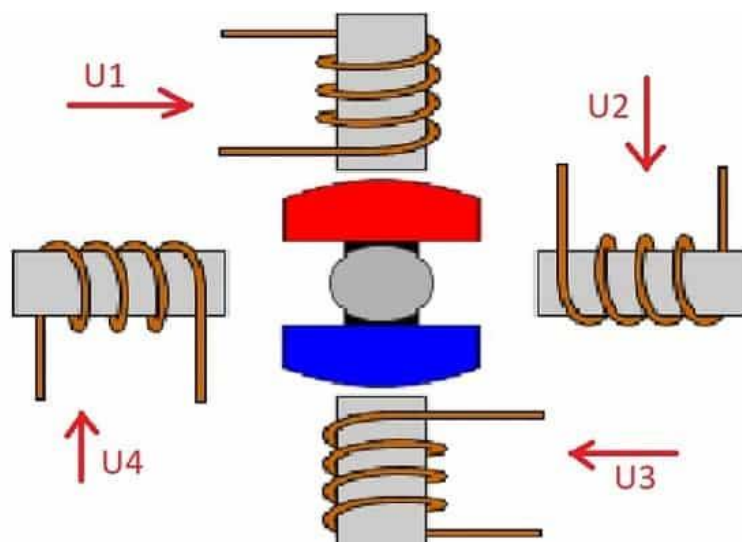


Рисунок 1.2 – Принцип действия шагового двигателя

Когда напряжение U_1 подается на первую обмотку, ротор перемещается на 90° ; когда U_2 , U_3 и U_4 поочередно подаются на соответствующие обмотки, вал продолжает вращаться, пока не совершит один полный оборот. Затем цикл повторяется снова. Чтобы изменить направление вращения, достаточно изменить порядок подачи импульсов на соответствующие обмотки.

Для обеспечения различных рабочих параметров важны, как величина шага для перемещения вала, так и крутящий момент, прикладываемый для перемещения вала. Изменение этих параметров достигается за счет конструкции самого ротора, способа подключения и конструкции обмоток.

Ротор магнитно взаимодействует с электромагнитным полем статора. Поэтому его конструкция и технические характеристики напрямую определяют режим работы и параметры вращения шагового устройства. Чтобы определить тип шагового двигателя на практике, необходимо вращать вал при обесточенной сети; если ощущается сопротивление, это говорит о наличии магнита, в противном случае конструкция не имеет магнитного сопротивления.

Реактивные шаговые двигатели (рисунок 1.3) не имеют магнитов в роторе и изготавливаются из магнитомягкого сплава. В поперечном сечении они напоминают шестеренку с зубьями. Полюса обмотки статора намагничиваются противоположными парами, создавая магнитную силу, которая перемещает ротор. Ротор перемещается под действием переменного тока, протекающего по парам обмоток.

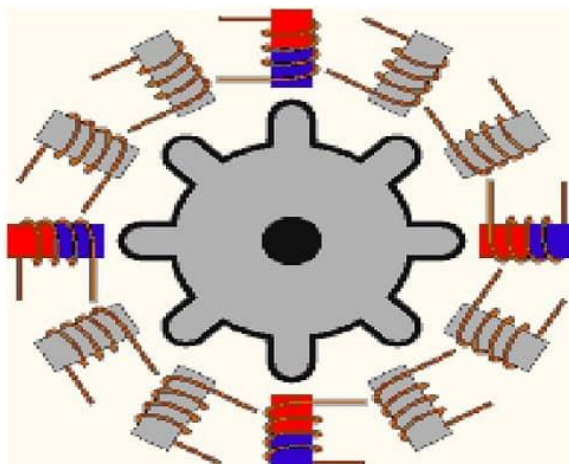


Рисунок 1.3 – С переменным магнитным сопротивлением

Основным преимуществом данной конструкции шагового привода является отсутствие тормозящего момента, создаваемого полем магнитом на выходе. Фактически, он идентичен синхронному двигателю, где ротор вращается в соответствии с магнитом поля статора. Недостатком является уменьшение величины крутящего момента. Шаг реактивных двигателей составляет от 5° до 15° .

В случае с постоянными магнитами (рисунок 1.4) подвижные элементы шагового двигателя собираются из постоянных магнитов и могут иметь два или более полюсов. Вращение ротора происходит за счет притяжения или отталкивания полюсов под действием электрического поля при подаче напряжения на соответствующую обмотку. В данной конструкции угловой шаг составляет от 45° до 90° .

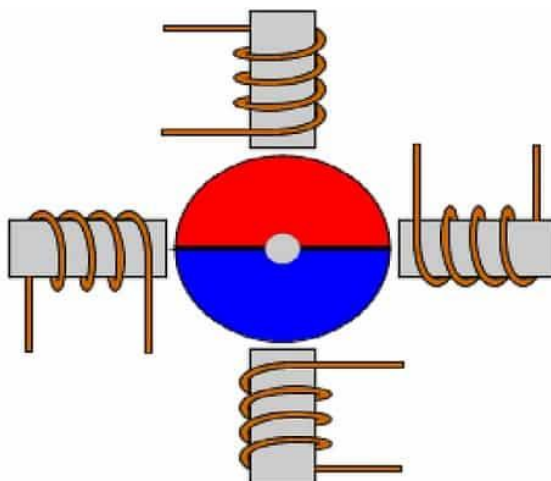


Рисунок 1.4 – С постоянным магнитом

Гибридная модель (рисунок 1.5) разработана таким образом, чтобы объединить преимущества двух предыдущих моделей при меньшем угле наклона и шаге. Ротор представляет собой цилиндрический постоянный магнит, намагниченный вдоль продольной оси. Конструктивно он выглядит как два круглых полюса, на поверхности которых расположены зубцы ротора из магнитного мягкого материала. Такое решение позволило добиться превосходной силы удержания и крутящего момента.

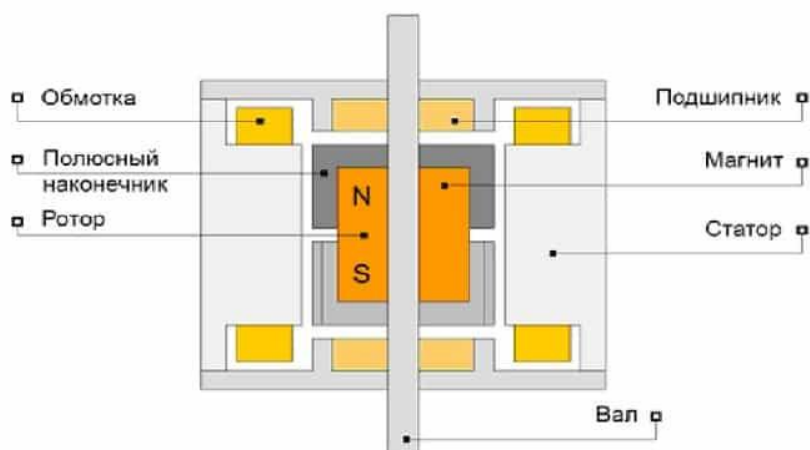


Рисунок 1.5 – Гибридная модель

Преимущества гибридных шаговых двигателей – высокая точность, плавность и высокая скорость в диапазоне от 0,9 до 5°. Они используются в высококласных станках с ЧПУ, компьютерной и офисной технике, а также в современной робототехнике. Единственным недостатком является их относительная дороговизна.

На практике шаговые двигатели являются полифазными. Плавность работы напрямую зависит от количества витков; чем больше витков, тем плавнее вращение, но и выше стоимость. В этом случае крутящий момент не увеличивается с ростом числа фаз, но их минимальное количество на статоре двигателя должно быть не менее двух. Количество фаз не определяет количество обмоток, поэтому двухфазный шаговый двигатель может иметь четыре и более обмоток.

Однополюсные шаговые двигатели (рисунок 1.6) характеризуются наличием ответвления от средней точки в схеме соединения обмоток. Это позволяет легко менять магнитные полюса. Недостатком такой конструкции является то, что используется только половина имеющихся обмоток, что приводит к низкому крутящему моменту. Поэтому она характеризуется большими размерами.

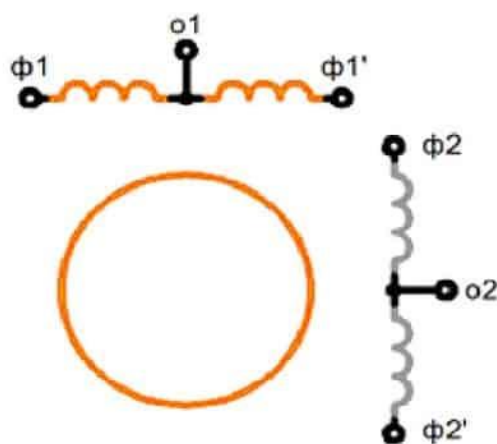
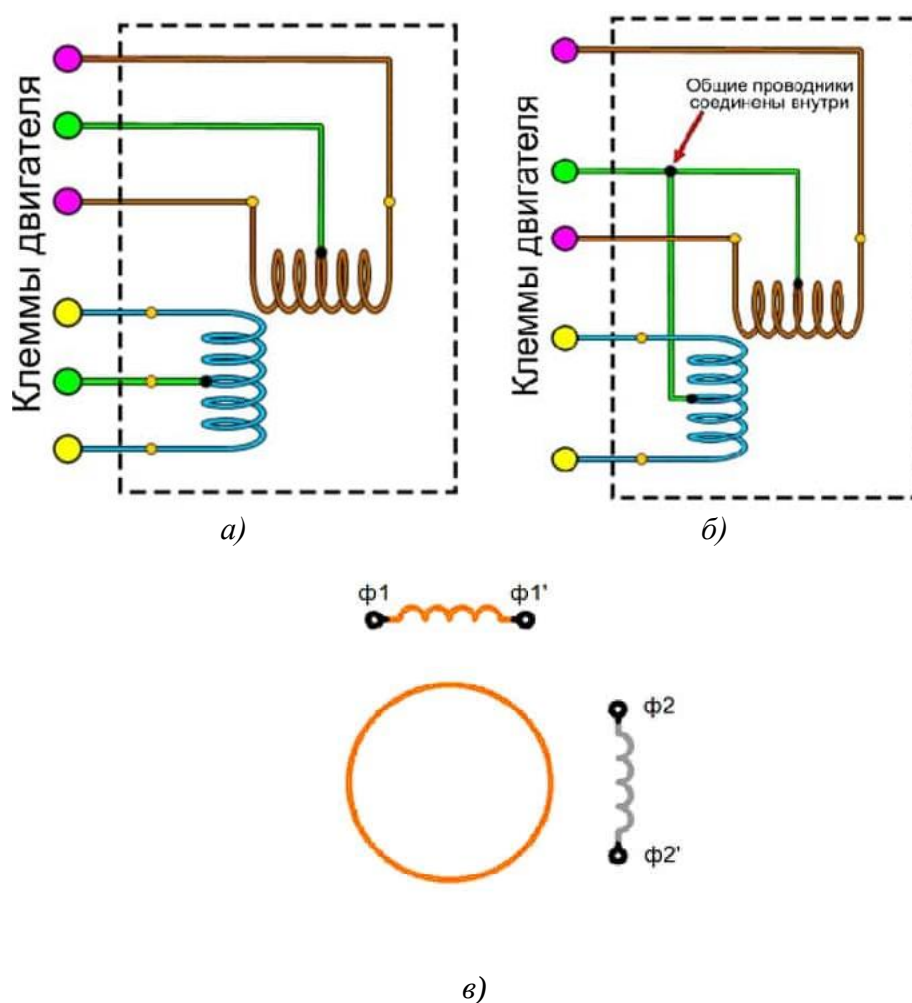


Рисунок 1.6 – Униполярный шаговый двигатель

Чтобы использовать всю мощность катушки, средний вывод следует оставить неподключенным. В однополярных конструкциях имеется пять или шесть выводов. Количество выводов зависит от того, берется ли средний вывод отдельно от каждой обмотки двигателя или соединяется вместе.

Биполярный шаговый двигатель подключается к контроллеру через четыре клеммы. Обмотки могут быть соединены последовательно или параллельно. Рассмотрим пример работы, показанный на рисунке 1.7.



a – с различными выводами; *б* – с одним выводом; *в* – двигатель

Рисунок 1.7 – Биполярный шаговый двигатель

На схеме такого двигателя видно, что в каждой фазе имеется по одной обмотке возбуждения. Поэтому изменение направления тока требует

использования в электронной схеме специального драйвера (электронной микросхемы, предназначенной для управления); аналогичный эффект может дать и включение Н-моста. По сравнению с предыдущим вариантом, биполярные устройства обеспечивают тот же крутящий момент при гораздо меньших размерах.

Для питания обмоток требуется устройство, способное выдавать серию управляемых импульсов или импульсов в постоянной последовательности. К таким устройствам относятся полупроводниковые приборы и микропроцессорные драйверы для подключения шаговых двигателей. Имеется набор выходных клемм, каждая из которых имеет свой способ питания и режим работы.

В зависимости от способа подключения необходимо использовать одну из клемм шагового блока. Различные варианты подключения тех или иных клемм к выходному сигналу постоянного тока позволяют получить определенные скорости вращения, шаги или микрошаги линейного перемещения в плоскости. Один и тот же двигатель может быть параметризован за счет драйвера, так как для одних задач требуется более низкая частота, а для других – более высокая.

Шаговые устройства могут управляться несколькими способами. Каждый из них имеет свой способ подачи сигнала на полюсные пары. Всего существует три способа активации обмоток.

Волна – в этом режиме возбуждается только одна обмотка, а полюса ротора притягиваются. В этом случае шаговый двигатель выдает только половину крутящего момента и не может тянуть большие нагрузки.

Полношаговый– в этом режиме фазы переключаются одновременно, т. е. возбуждаются обе фазы сразу. Это обеспечивает максимальный крутящий момент и, в случае параллельного или последовательного соединения обмоток, максимальное напряжение или ток.

Полушаговый– это комбинация двух предыдущих методов переключения обмоток. При реализации этого метода шаговый двигатель

подает переменное напряжение сначала на одну катушку, а затем на две. Это улучшает блокировку на максимальной скорости и увеличивает количество шагов.

Для более мягкого управления и преодоления инерции ротора используется микрошаговое управление, при котором синусоидальные сигналы передаются в виде микрошаговых импульсов. В результате силы взаимодействия в магнитной цепи шагового двигателя изменяются более плавно, что приводит к более плавному перемещению между полюсами ротора. Рывки шагового двигателя могут быть значительно уменьшены.

Для реализации данного проекта был выбран и проанализирован шаговый двигатель типа *FL28STH32-0956A* (*shaft* 5 мм). Это униполярный шаговый двигатель (2 обмотки, 4 вывода). Размеры – 28×32 мм, угловой шаг 1,8°, момент 430 г·см, напряжение от 5 до 10 В, вал 5 мм с одной стороны, сопротивление фазы 2,8 Ом, масса 0,11 г.

1.3 Типы температурных датчиков и принцип их работы

Основной принцип работы датчиков температуры в системах автоматического управления заключается в преобразовании температуры в электрические величины. Преимуществами датчиков температуры являются удобство высокоскоростной передачи данных на большие расстояния, возможность обратного преобразования, преобразования в цифровой код и чувствительность измеряемой величины. Существует несколько типов устройств.

Принцип действия приборов основан на термоэлектрическом эффекте: в замкнутой цепи из двух полупроводников или проводников разная температура в точке спая (месте контакта) порождает там ток. Переход в среде, температура которой измеряется, называется «горячим», а переход на противоположной стороне – «холодным». Чем больше температура измеряемой среды отличается от температуры воздуха, тем больше ток. Некоторые из этих измерительных приборов имеют изолирующий слой, а

некоторые – нет. В последнем случае термопары можно использовать только в цепях, не имеющих контакта с землей.

Хромель-алюмель. В основном используется в промышленности. Особенности: широкий диапазон измеряемых температурот минус 200 до 13000 °С, доступная цена. Не может использоваться в цехах с высоким содержанием серы.

Хромель-копель. По применению аналогичен предыдущему типу, но работает только в неагрессивных жидких или газообразных средах. Часто используется для измерения температуры в мартеновских печах.

Железо-константовые. Действует в разбавленных атмосферах.

Платинородий-платиновые. Самые дорогие. Отличается стабильностью и точностью показаний. Используется для высокотемпературных измерений.

Вольфрам-рениевые. Обычно с защитным покрытием. Основное применение – измерение очень высокотемпературных сред.

Принцип действия термодатчиков сопротивления (ТДС) основан на температурной зависимости сопротивления проводников или полупроводников [5]. Проводники имеют высокий температурный коэффициент сопротивления и изготавливаются из материалов с линейной зависимостью сопротивления от температуры. Вышеперечисленные свойства применимы к пластинам и в несколько меньшей степени к меди.

Преимущества проводниковых термометров сопротивления:

- простая и надежная конструкция, что делает их пригодными для использования в машиностроении и электронике;
- высокая точность и чувствительность;
- простое устройство для считывания показаний.

Полупроводниковые датчики температуры имеют высокостабильные характеристики с течением времени. Полупроводниковые термисторы имеют высокий температурный коэффициент сопротивления (ТКС). Датчики температуры с отрицательным ТКС называются термисторами (сопротивление уменьшается с ростом температуры), а датчики температуры

с положительным ТКС–позисторами (сопротивление увеличивается с ростом температуры). Термисторы называются *NTC*, а позисторы–*PTC*.

Аналоговые термометры обычно недороги и не требуют сложного обслуживания. Основная проблема – это шкала. Они показывают температуру с высокой точностью, но либо имеют очень маленькие интервалы измерений, либо охватывают широкий диапазон температур, но точность при этом приблизительная.

Цифровые дороже аналоговых систем, но гораздо точнее. Способны измерять в широком диапазоне и используются в быту и в техническом секторе.

Конструктивные элементы цифровых термометров:

- чувствительный элемент (обычно термистор);
- аналогово-цифровой преобразователь: преобразует электрический сигнал от термистора в цифровой;
- дисплей;
- элемент питания;
- сигнальные входы и выходы, необходимые для взаимодействия с другими устройствами.

Для реализации данного проекта был выбран накладной резистивный датчик температуры типа *Pt 100*. Чувствительный элемент *Pt 100* со стандартной характеристикой – канальный датчик совместим с любыми контроллерами (*PLC*); зонд из нержавеющей стали *SS 304* и термостойкая изоляция проводов – работает при температуре до плюс 240 °С, можно использовать как датчик дымовых газов. Измерение от минус 50 до плюс 240 °С – используется во всех регионах России, в том числе с экстремально холодным климатом и в промышленности; чувствительный элемент точности класса *A* или *B* в зависимости от исполнения.

1.4 Газодинамические датчики потока

Принцип действия газодинамических датчиков основан на эффекте перепада давления в потоке газа, проходящего через локальное сужение. Поток вокруг этого сужения создает перепад давления, который преобразует значение скорости потока в значение перепада давления. Перепад давления измеряется высокочувствительными датчиками. В общем случае процесс измерения скорости потока включает в себя двухступенчатую схему преобразования. Преобразование скорости потока в дифференциальное давление является нелинейной функцией, а также зависит от удельного веса и давления газа, что может привести к ошибкам измерения [6]. Датчик состоит из пневматической и электрической частей. Пневматическая часть может быть подключена к устройству с помощью достаточно длинной линии. Электрическая часть датчика обычно устанавливается в корпус прибора для дополнительной защиты.

Основные преимущества газодинамических датчиков:

- датчики очень просты и надежны;
- пневматическая часть датчика недорогая и одноразовая;
- датчик двусторонний и симметричный для разных направлений потока;
- высокая воспроизводимость характеристик;
- малый вес и габариты.

Основные недостатки газодинамических датчиков:

- влияние на измеряемый расход газа;
- возможность получения неверных значений при конденсации водяного пара в измерительной линии;
- нелинейность передаточных характеристик;
- зависимость от удельной плотности и давления анализируемого газа.

Несмотря на очевидные недостатки, датчик широко используется в вентиляторах и спирографах. Благодаря простой конструкции и низкой стоимости, это единственный датчик расхода, выпускаемый в одноразовом виде, что немаловажно в современных условиях. Конструкция датчика позволяет использовать его для любого типа санитарной обработки. При непрерывном производстве гарантируется высокая точность работы датчика, поэтому при замене датчиков не требуется калибровка оборудования.

Датчики расхода, основанные на эффекте теплопроводности. Этот метод основан на принципе регистрации изменения температуры нагретого тела после контакта со струей газа. Существует несколько усовершенствованных методов измерения скорости потока газа, использующих эффект охлаждения нагретого чувствительного элемента и эффект теплопередачи от нагревательного элемента к чувствительному элементу. Нагревательные элементы могут быть расположены таким образом, чтобы линии потока были разными в зависимости от расхода. В любом случае разница температур пропорциональна скорости потока, удельной теплоемкости и температуре газа. В данном случае скорость потока пропорциональна разнице между измеренными значениями температуры первого и второго датчиков.

Основные преимущества датчиков термического сопротивления:

- высокая чувствительность и точность;
- небольшой вес и габариты;
- датчики двухсторонние и симметричные для разных направлений потока.

–

Основные недостатки датчиков термического сопротивления.

- зависимость от удельной теплоемкости и температуры газа;
- влияние на измеряемый расход;

- тонкий чувствительный элемент легко повреждается при дезинфекции;
- необходимость калибровки при замене датчика.

Данный тип датчиков используется в аппаратах искусственной вентиляции легких и может устанавливаться непосредственно на эндотрахеальную трубку благодаря небольшому весу и повышенной чувствительности.

Для реализации проекта был выбран датчик потока воздуха типа EE650. Рабочий диапазон от 0 до 10/15/20 м/с, выходной сигнал от 4 до 20 мА или от 0 до 10 В, так же, как и время отклика, 1 или 4 секунды, можно выбрать с помощью джамперов.

1.5 Типы датчиков влажности и принцип их работы

Гигрометр, также известный как датчик влажности, это устройство, применяемое для определения уровня влажности. Этот параметр имеет существенное значение в повседневной жизни, а также для функционирования разнообразных устройств, в аграрном секторе (например, для измерения влажности почвы) и в других областях.

На производстве влажность влияет на качество и сохранность продукции и оборудования, а в аграрной отрасли — на урожайность и другие аспекты. Для мониторинга влажности используется гигрометр. [7].

Измерительные приборы могут быть предварительно откалиброваны на определенные значения, однако для более точной калибровки необходимо знать точный уровень влажности окружающей среды. Влажность может измеряться в различных единицах: для абсолютной влажности используют граммы на кубический метр, а для относительной — проценты RH. При определении влажности твердых или жидких материалов уместно использовать массовые проценты от общего веса образца. В случае несмешивающихся жидкостей применяют единицу измерения ppm, указывающую на количество частей воды на миллион частей веса образца.

Существуют различные типы гигрометров, классифицируемые по принципу их работы:

- емкостные;
- резистивные;
- термисторные;
- оптические;
- электронные.

Емкостные гигрометры работают по простому принципу: воздушный зазор служит диэлектриком для датчика. Поскольку диэлектрические свойства воздуха изменяются в зависимости от влажности, колебания влажности влияют на емкость воздушного конденсатора (рисунок 1.8).



Рисунок 1.8 – Конструкция емкостного датчика влажности

Более сложные емкостные датчики с воздушным зазором включают диэлектрик, чья диэлектрическая проницаемость значительно меняется под воздействием влажности, что улучшает характеристики датчика по сравнению с базовой версией с воздушным зазором (рисунок 1.9).



Рисунок 1.9 – Емкостной датчик влажности

Для измерения влажности в твердых материалах используется другой подход: между пластинами конденсатора помещается объект, например, таблетка (рисунок 1.10). Конденсатор подключается к колебательному контуру и электронному осциллятору, после чего измеряется собственная частота контура, и на основе этих данных рассчитывается емкость, возникающая при введении объекта в контур.

Однако для достижения высокой точности измерений необходимо соблюдение определенных условий, таких как низкое содержание влажности в объекте (менее 0,5%) и отсутствие частиц с высокой диэлектрической проницаемостью. Также важно, чтобы форма объекта оставалась неизменной в течение всего процесса измерения.

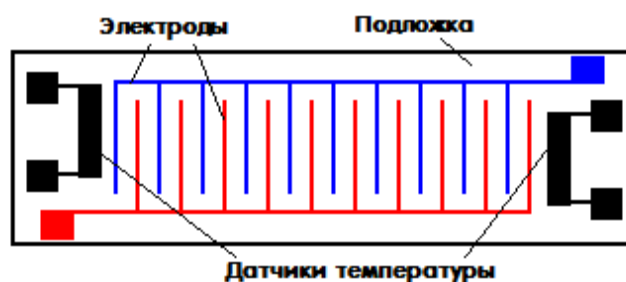


Рисунок 1.10 – Емкостной тонкопленочный датчика влажности

Емкостной тонкопленочный гигрометр - это третий тип емкостного датчика влажности (рисунок 1.11), который имеет два гребенчатых электрода

на подложке, выступающие в качестве оболочки. Для компенсации температуры этот датчик также оснащен двумя дополнительными температурными датчиками.

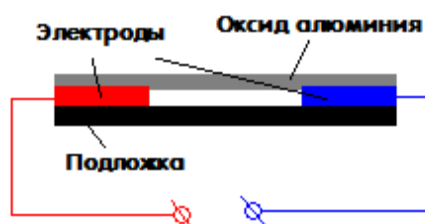


Рисунок 1.11 – Конструкция резистивного датчика влажности

Эти датчики состоят из двух электродов, прикрепленных к подложке, под которыми находится слой материала с низким сопротивлением, изменяющимся в зависимости от влажности. Одним из подходящих материалов для таких устройств является оксид алюминия, который хорошо впитывает влагу из окружающей среды, изменяя свое удельное сопротивление. Это приводит к изменению сопротивления измерительной цепи в зависимости от влажности, что позволяет определять уровень влажности через протекающий ток. Благодаря их низкой стоимости, эти датчики имеют преимущество.

Термисторные гигрометры (рисунок 1.12) состоят из двух одинаковых термисторов, которые являются нелинейными электронными компонентами, сопротивление которых сильно зависит от температуры. Один из термисторов помещается в герметичную камеру с сухим воздухом, а другой помещается в камеру с влажным воздухом. Подключенные к мостовой схеме, термисторы позволяют определять влажность по значению напряжения, полученного в результате измерений, позволяя определить разницу во влажности в воздухе. Это происходит потому, что вода испаряется из корпуса термистора при повышении влажности воздуха, что приводит к его охлаждению и изменению температуры.

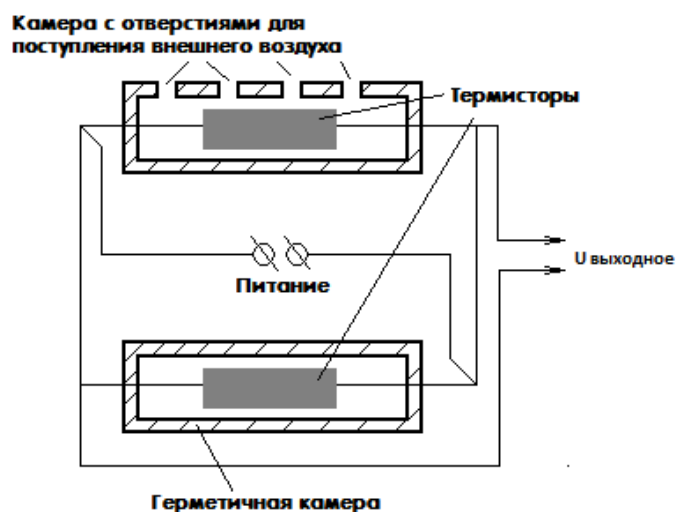


Рисунок 1.12 – Конструкция термисторного датчика влажности

Оптические (конденсационные) гигрометры (рисунок 1.13) являются самым точным типом датчиков влажности. Их принцип работы основан на концепции "точки росы", при достижении которой газовая и жидкая фазы вступают в равновесие. Это означает, что при определенной температуре на поверхности предмета начнет собираться конденсат, переходя в жидкую фазу. Точка росы тесно связана с параметрами окружающей среды, такими как влажность и давление, и при измерении этих параметров можно точно определить влажность воздуха.



Рисунок 1.13 – Оптический датчик влажности

Принцип работы оптических датчиков влажности заключается в освещении зеркала светодиодами, его охлаждении, а затем наблюдении за конденсацией влаги и реакцией фотодетектора. Эти гигрометры обладают высокой точностью и отсутствием гистерезиса, но требуют больше энергии и

Электронные датчики влажности (рисунок 1.14) работают на основе изменений концентрации электролита на поверхности изоляционных материалов, причем некоторые автоматически регулируются в зависимости от точки росы. Для максимальной точности иногда используется концентрированный раствор хлорида лития, который является чувствительным к изменениям влажности. Гигрометры могут дополняться термометрами для удобства использования.

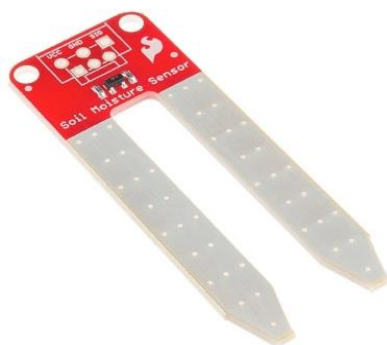


Рисунок 1.14 – Электронный гигрометр

Этот прибор обладает высокой точностью и малой погрешностью. Он может измерять влажность независимо от температуры окружающей среды.

Для реализации данного проекта был выбран датчик влажности типа *НТР 201*. Измеряемые значения *RH* доступны на аналоговых выходах: напряжением от 0 до 10 В или током от 4 до 20 мА. Кроме того, датчик *НТР 201* имеет широкий диапазон напряжения питания.

2 Разработка системы автоматизированного управления газодинамическим стендом

2.1 Микропроцессорная техника в системах управления технологическими процессами

Микропроцессорная техника (МТ) – это поколение управляющего оборудования, содержащего микропроцессорные интегральные схемы.

Микропроцессорная техника включает в себя аппаратные устройства и программные продукты.

Технические средства предназначены для первичной обработки технической информации, собираемой датчиками, отображения информации для обслуживающего персонала, обработки и формирования исполнительных механизмов и выдачи управляющих воздействий [8].

Микропроцессор (МП) был изобретен в 1969 году, а в 1970-х годах появился новый класс ЭВМ-микрокомпьютеры. Наличие микропроцессора изначально было определяющим признаком микрокомпьютера. Сейчас микропроцессоры используются во всех без исключения классах компьютеров.

Различают универсальные (многопользовательские – мощные микрокомпьютеры, однопользовательские или персональные) и специальные (многопользовательские – серверные, однопользовательские – рабочие станции) микрокомпьютеры.

Многопользовательские микрокомпьютеры – это мощные микрокомпьютеры с несколькими видеотерминалами, функционирующие в режиме разделения времени и позволяющие эффективно работать нескольким пользователям.

Основными характеристиками микрокомпьютеров являются быстродействие, производительность и тактовая частота.

Модульный принцип построения вычислительной техники позволяет создавать семейства компьютеров с различными функциональными назначениями, техническими характеристиками и дизайном.

Для локального управления техническими процессами (ТП) обычно используются однокристалльные компьютеры со встроенными АЦП и ЦАП. Они обычно выполняются на одной БИС (крупногабаритной интегральной схеме).

Микропроцессоры и микроконтроллеры – это гибкие, программно управляемые, многофункциональные логические устройства, которые могут принимать, хранить, обрабатывать (логически или арифметически) и выдавать информацию, представленную в виде многоразрядного двоичного кода.

Микропроцессоры по своим логическим функциям и структуре похожи на процессоры в обычных компьютерах, с 2, 16, 32, 64, 128 битами. Основная функция МП – преобразование информации. Однако сам МП не может обрабатывать информацию, то есть решать конкретные задачи; он должен быть подключен и запрограммирован к устройствам памяти (УП) и устройствам ввода-вывода (УВВ), и обмен информацией МП с этими устройствами должен быть гарантирован.

Программируемость – возможность изменения алгоритмов обработки данных путем программирования. Программируемость позволяет применять один тип микропроцессорного устройства для решения различных задач управления.

Программируемый логический контроллер (ПЛК) – это микропроцессорное устройство, взаимодействующее с оператором, выполняющее функции логического анализа и управления по соответствующим алгоритмам в соответствии с требованиями технического процесса.

Микропроцессорная система (МПС) – это набор взаимосвязанных устройств, состоящий из одного или нескольких устройств памяти, устройств

ввода / вывода и ряда других устройств, обеспечивающих выполнение определенных функций.

В МПС существует четыре основных архитектурных элемента: устройства управления (УУ), арифметико-логические блоки (АЛУ), устройства памяти (УП) и устройства ввода-вывода (УВВ).

АЛУ выполняет арифметические и логические операции (сложение, вычитание, сдвиг, перенос, логическое сложение ИЛИ, логическое умножение И и сложение по модулю) над данными, поступающими от устройств памяти и устройств ввода / вывода, в соответствии с указаниями управляющего устройства.

УУ расшифровывает команды, записанные в программе, и генерирует управляющие сигналы, необходимые АЛУ и всей системе в целом для выполнения требуемых функций; АЛУ и УУ обычно построены как одна большая интегральная схема (ИС), называемая центральным процессором (ЦП).

Персональные компьютеры (ПК) – это однопользовательские микрокомпьютеры, отвечающие требованиям доступности для населения и универсальности применения.

Рабочие станции – мощные однопользовательские микрокомпьютеры, специализирующиеся на выполнении определенных типов задач.

Серверы – это специализированные многопользовательские мощные микрокомпьютеры, которые обрабатывают запросы всех станций в компьютерной сети.

УВВ обеспечивают связь между процессором и внешним миром (периферийными устройствами).

УП (запоминающее устройство или накопитель) – это место для хранения программ и данных, закодированных в двоичной форме. УП, входящий в состав МП, представляет собой электронное устройство, состоящее из элементов памяти (ячеек). Каждой ячейке присваивается номер,

называемый адресом. Устройства ввода-вывода обеспечивают связь с внешними (периферийными) устройствами.

Информация в МП вводится и хранится по принципу «потенциал высокого уровня – потенциал низкого уровня», что соответствует 1 или 0 в двоичной системе счисления. Такая форма представления информации (программы) называется машинным кодом; одна цифра двоичного числа называется битом, который является наименьшим количеством информации; запись десятичных чисел или букв в машинном коде называется кодированием.

Основной логической единицей информации в МП является слово, которое представляет собой набор двоичных чисел, обрабатываемых одновременно. Количество двоичных цифр, составляющих слово, определяется длиной слова. Длина слова характеризует количество кодовых комбинаций и количество адресов, по которым может обращаться УВВ. Слово длиной 8 бит называется байтом. Общее количество ячеек в УП определяет объем памяти.

Поскольку МП не имеет устройств ввода / вывода, памяти, панели управления или источника питания, сам МП не может быть использован для управления и измерения или выполнения функций управления. Выполнение предоставленных пользователем функций по управлению техническими процессами или сбору и обработке информации осуществляется МП-системой, которая оснащена перечисленными блоками и устройствами для взаимодействия с ними, называемыми интерфейсами.

Понятие интерфейса охватывает набор аппаратных средств и программ, выполняющих функции сопряжения отдельных блоков и обмена информацией между ними; в МП-системах различают внутренние и внешние интерфейсы.

Внутренние интерфейсы – это линии связи или шины, обеспечивающие взаимосвязь внутренних элементов МП. Внешние интерфейсы обеспечивают

взаимосвязь и обмен информацией между МП, устройствами памяти, УВВ и внешними устройствами (например, датчиками технических параметров).

Стандарты аппаратных интерфейсов в информатике называются протоколами. Протокол – это набор технических решений, которые разработчик устройства должен обеспечить для успешного взаимодействия с другими устройствами.

Для связи между элементами в МП-системе используются три шины: шина адреса (ША), шина данных (ШД) и шина управления (ШУ). Физически шины выполнены в виде печатных проводников или кабелей, к которым подключаются все блоки системы (рисунок 2.1).

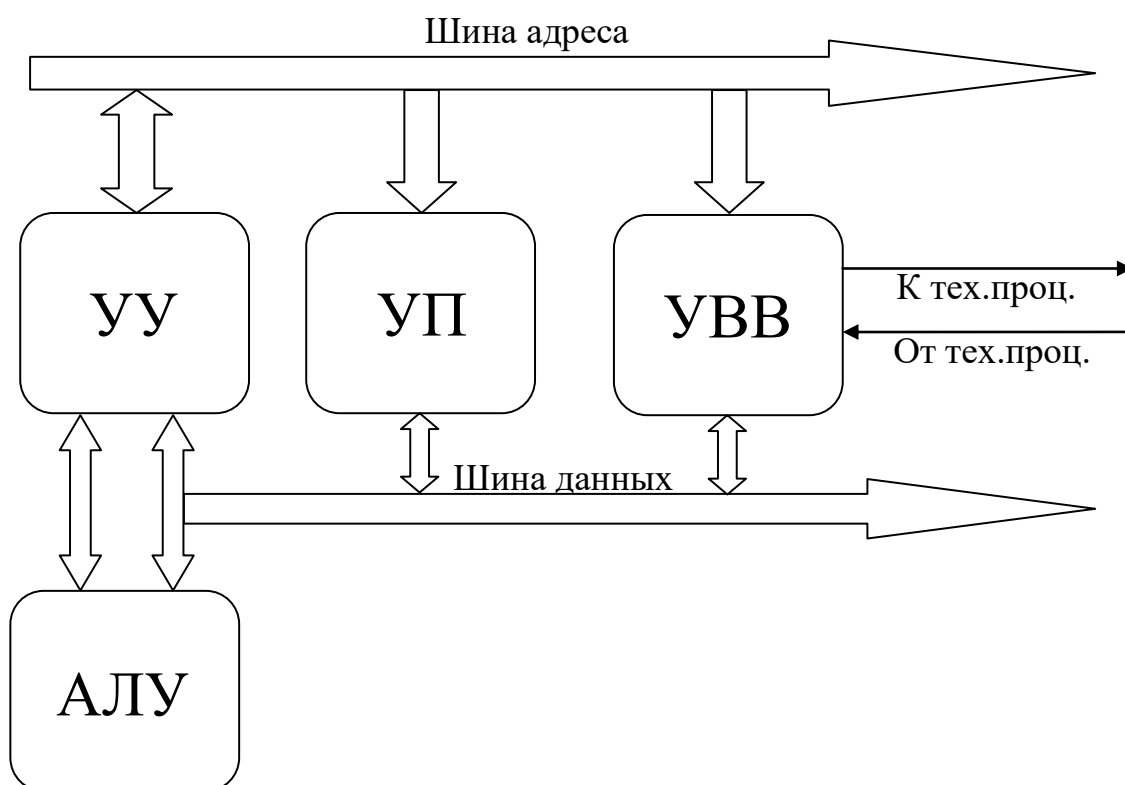


Рисунок 2.1 – Схема микропроцессорной системы

Однонаправленная (односторонняя) шина позволяет управляющему устройству выбрать любое слово или любое устройство ввода-вывода в

памяти, соответствующее комбинации нулей и единиц (двоичный адрес) на шине.

Двунаправленная (действующая в любом направлении) шина позволяет передавать информацию как от устройств ввода в память, так и из памяти в АЛУ или устройство вывода.

ШУ используются только для вывода сигналов и являются однонаправленными.

Центральные процессоры, обменивающиеся информацией с внешними устройствами, не находятся в непосредственном контакте с ША или ШД. Надежная передача информации осуществляется с помощью временного хранения (фиксированного, «защелкивающегося») с использованием буферных регистров. Буфер – это устройство, соединяющее две разнородные системы для согласования их характеристик и совместной работы.

В МП-системе имеется множество внешних устройств, и для координации их работы вводится устройство, называемое контроллером ввода-вывода, которое определяет необходимый источник входной информации и приемник ее вывода. Буферные регистры и контроллеры образуют порты ввода и вывода соответственно.

Порты – это устройства ввода / вывода для связи процессора с внешними устройствами. В качестве портов ввода / вывода используются специально программируемые БИС.

Архитектура МП была разработана с учетом требований гибкости МП, высокой производительности и низкой стоимости. Гибкость необходима для создания эффективного математического обеспечения и расширения функциональных возможностей при создании систем управления; быстродействие МП достигается за счет параллельной обработки нескольких разрядов информационных слов в АЛУ и использования быстродействующих элементных баз. Экономия средств обеспечивается за счет однородности схемы, минимального количества контактов и внешних выводов на кристалле МП. Современные компьютеры и вычислительные

многофункциональные машины используют блочно-модульную конструкцию, когда аппаратные конфигурации, необходимые для выполнения конкретных типов задач, могут быть собраны из готовых узлов и блоков.

Любой компьютер представляет собой неразрывное соединение программного и аппаратного обеспечения. Программа – это упорядоченная последовательность команд. Программная конфигурация компьютерной системы называется конфигурацией программного обеспечения. Так же как существуют взаимосвязи между физическими узлами и блоками, существуют взаимосвязи между программами. Базовый уровень – это самый низкий уровень программного обеспечения. Базовое программное обеспечение отвечает за взаимодействие с базовым оборудованием. Оно записывается в программируемую микросхему памяти. Программы системного уровня обеспечивают взаимодействие с другими программами в компьютерной системе и с программами базового уровня. Программа, отвечающая за взаимодействие с конкретным устройством, называется драйвером устройства и является частью программного обеспечения системного уровня. Другой класс программ системного уровня отвечает за взаимодействие с пользователем. Эти программные средства называются средствами пользовательского интерфейса. Они напрямую влияют на удобство работы на рабочей станции.

Совокупность программ системного уровня составляет ядро операционной системы компьютера и является необходимым условием для обеспечения практической работы человека с компьютерными системами.

Программное обеспечение прикладного уровня – это набор прикладных программ, с помощью которых на конкретном рабочем месте выполняются определенные задачи.

2.2 Язык программирования

Работа системы МП организована в соответствии с заранее заданной программой, состоящей из последовательности инструкций, которые выполняются шаг за шагом. Шаги в этой последовательности называются инструкциями. Программы управления процессом могут быть написаны в машинном коде с использованием двоичной системы. Сначала программа пишется на символьном (входном) языке, который затем переводится на машинный язык вручную или с помощью компьютера. Входной язык состоит из фиксированного набора команд. Мнемокод – самый простой из входных языков [8]. Операторы в мнемокоде представляют собой условные сокращения названий операций в естественных языках, что позволяет легко запоминать и использовать команды программы. Мнемокод ускоряет программирование и является основой для создания более сложных алгоритмических языков. Мнемокоды могут использоваться как для кодирования адресов, так и команд. Символы (метки) используются для описания программы и ее отдельных частей. В качестве метки целесообразно использовать символическое обозначение (адрес) части программы, например, «*START*», «*DELAY*» и т. д. Метка помещается перед кодом операции, за которым следует двоеточие. При переводе программы каждой метке присваивается текущее значение счетчика команд, что фиксирует определенное место в программе и облегчает переход к нужной части программы или многократное повторение отдельной части.

Язык программирования ассемблер широко используется для технических объектов управления. Его основу составляют мнемоники команд МП-машины и специальные команды, управляющие преобразованием исходной программы в язык машины. Эти команды называются директивами ассемблера или псевдокомандами. Директивы ассемблера содержат информацию о начале и конце программы и другую информацию, необходимую для формирования и размещения данных в памяти. Для обозначения директив также используются мнемонические

сокращения. Метка директивы должна содержать имя символа, исключая двоеточие. Наиболее распространенными директивами ассемблера являются «*APPEND*» (знак равенства) и «*MASSIVE*» (таблица). Первая директива задает начальные числовые значения для всех имен условий, а вторая определяет упорядоченную последовательность чисел.

Перевод программы с ассемблера на машинный язык может быть осуществлен вручную с помощью специального кодирующего устройства или программно с помощью компьютера и специальной программы (транслятора). Чтобы написать программу, ее необходимо распечатать или перечислить в соответствии с форматом, требуемым для машинного перевода. Широко используются языки высокого уровня *VisualBasic* и *VisualBasicforApplications*, *C*, *C++* и языки технического программирования *IEC 1131-3 (SFC, FBD, LD, IL, ST)*.

2.3 Микроконтроллеры серии 8051

Рассмотрим основные функциональные узлы типичного микроконтроллера 8051/8052 и принцип их работы (рисунок 2.2). Также кратко опишем системные инструкции микроконтроллеры серии 8051 [9].

В микроконтроллере 8051 все вычисления производятся в арифметическом логическом устройстве, которое является частью базового микропроцессорного модуля (МПМ). Обмен данными, хранящимися в оперативной памяти микроконтроллера, и командами чтения осуществляется по внутренней шине 8051; по этой же шине происходит обмен данными с портами ввода / вывода (*P1 – P3*), последовательными портами и таймерами. Контроллер внутренней шины генерирует сигналы (*EA, ALE, PSEN, RD/WR*), необходимые для работы с внешней памятью программ и данных, а также сигнал сброса / установки *RST*.

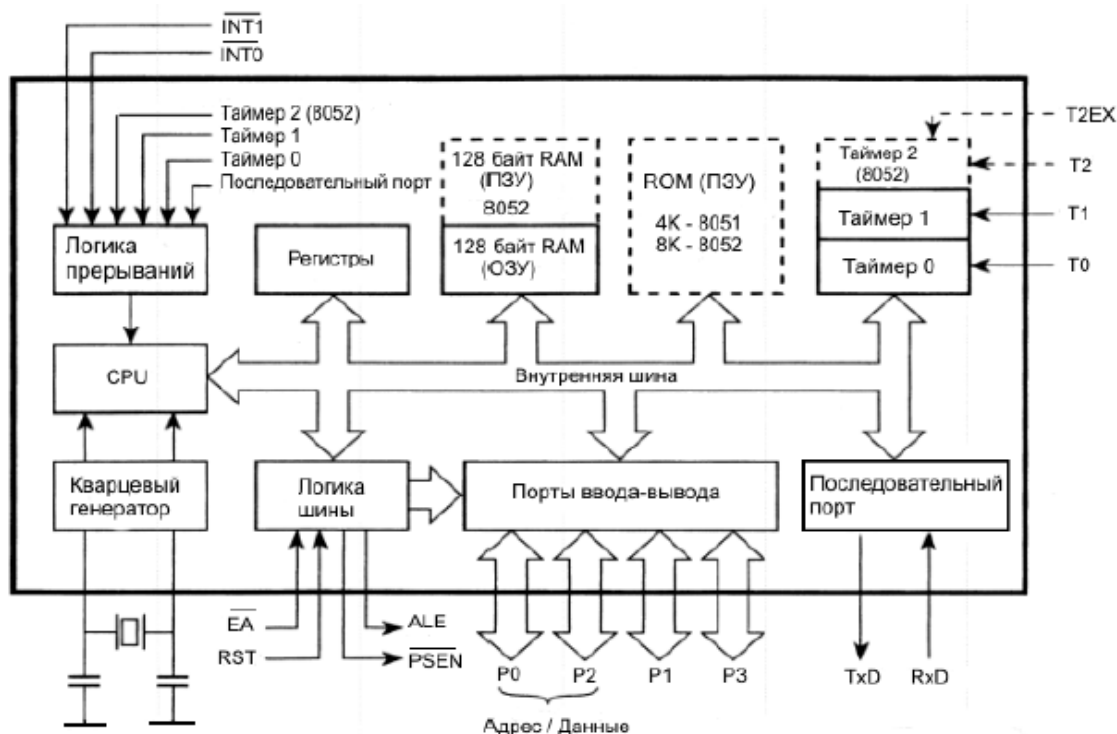


Рисунок 2.2 – Аппаратная архитектура микроконтроллера серии 8051

Микроконтроллер 8051 предназначен для работы в системах реального времени, которые могут генерировать определенные сигналы, требующие немедленной реакции микроконтроллера. Для обработки таких сигналов (или событий) используется логика прерываний, реализованная аппаратно, для обработки сигналов от внешних источников, таймеров и последовательных портов.

Скорость работы систем на базе 8051 варьируется от единиц до десятков мегагерц, в зависимости от тактовой частоты, на которой работает кристалл. В архитектуру классического микроконтроллера 8051 было внесено несколько изменений (к двум существующим таймерам был добавлен третий таймер и расширена внутренняя память), в результате чего появились наиболее популярные сегодня устройства 8052.

Микроконтроллер 8051 реализован в виде однокристалльного устройства, внешние выводы которого промаркированы, как показано на рисунке 2.3.

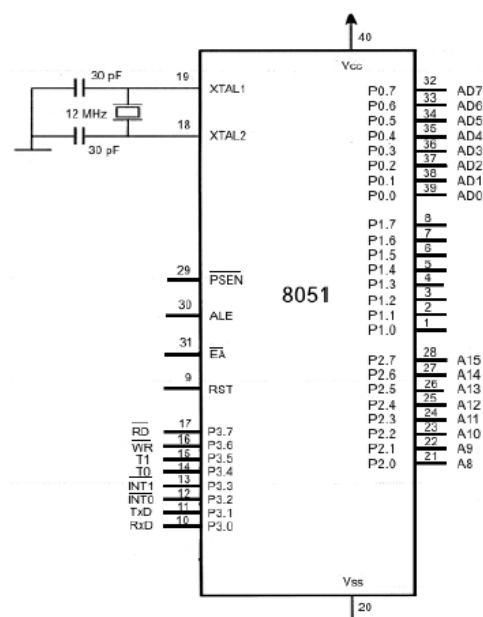


Рисунок 2.3 – Микроконтроллер серии 8051

Входные и выходные сигналы микроконтроллера 8051 имеют следующие назначения:

- *XTAL* и *XTAL2* – входы подключения кварцевого резонатора для работы генератора тактовой частоты микроконтроллера;
- *PSEN* – сигнал, используемый при обращении к внешней памяти программ;
- *ALE* – выходной сигнал разрешения фиксации адреса при обращении к внешней памяти программ / данных;
- *EA* – сигнал, блокирующий работу с внутренней памятью;
- *RST* – сигнал общего сброса;
- *P0* – *P3* – выходы портов ввода / вывода микроконтроллера;
- *Vss* и *Vcc* – выходы подачи напряжения питания.

Порты *P0*, *P2* и *P3* помимо функционирования в режиме ввода / вывода дискретных сигналов могут выполнять, в зависимости от аппаратной конфигурации, и другие функции. Так, через порт *P0* при обращениях к внешней памяти выставляются младшие 8 бит 16-разрядного адреса, а затем, в фазе записи / чтения данных, через этот порт идет обмен данными. Порт *P2*

приобретениях к внешней памяти служит источником старших 8 бит 16-разрядного адреса.

Выводы порта *P3* микроконтроллера 8051 имеют следующие альтернативные назначения:

- *P3.0* – вход приема данных в последовательный порт;
- *P3.1* – выход передачи данных с последовательного порта;
- *P3.2* – вход внешнего прерывания *INT0*;
- *P3.3* – вход внешнего прерывания *INT1*;
- *P3.4* – вход управления таймером 0;
- *P3.5* – вход управления таймером 1;
- *P3.6* – выход сигнала записи в память;
- *P3.7* – выход сигнала чтения из памяти.

Для использования альтернативных функций порта *P3* необходимо настроить соответствующим образом программное обеспечение системы 8051.

Микроконтроллеры семейства 8051 представляют собой микропроцессорные устройства с *CISC*-архитектурой и стандартным набором инструкций, характерным для этой архитектуры. Система инструкций 8051-совместимых устройств содержит 111 базовых инструкций размером от одного до трех байт, но большинство из них – одно- или двухбайтовые. Почти все инструкции выполняются за один или два машинных цикла, время выполнения которых составляет от 1 до 2 мкс при тактовой частоте 12 МГц, за исключением инструкций умножения и деления, для выполнения которых требуется четыре машинных цикла.

Команды микроконтроллера 8051 используют прямую, непосредственную, косвенную и неявную адресацию данных. Операндами команд могут быть отдельные биты, комбинации из четырех битов (тетрады), байты или двухбайтовые слова.

Набор инструкций 8051 содержит ряд инструкций, которые обеспечивают специфические функции управления, например, манипулирование отдельными битами; особенностью системы инструкций 8051 является возможность обращения к отдельным битам в памяти данных, а также к отдельным битам в специальных функциональных регистрах.

По выполняемым функциям команды микроконтроллера 8051 можно разделить на несколько групп:

- пересылки данных;
- арифметических операций;
- логических операций;
- операций над битами;
- передачи управления.

Рассмотрим эти группы команд более подробно, но перед этим условимся при описании мнемоники команд использовать следующие обозначения:

- Rn ($n = 0...7$) – регистр общего назначения в выбранном банке регистров;
- $@Ri$ ($i = 0, 1$) – регистр общего назначения в выбранном банке регистров, используемый для формирования косвенного адреса;
- $addr$ – адрес байта;
- src – адрес байта-источника;
- dst – адрес байта-приемника;
- $addr11$ – 11-разрядный абсолютный адрес перехода;
- $addr16$ – 16-разрядный абсолютный адрес перехода;
- $label$ – относительный адрес перехода;
- $\#direct8$ – непосредственный операнд размером 1 байт;
- $\#direct16$ – непосредственный операнд размером 2 байта;
- bit – адрес прямо адресуемого бита;

- $\sim bit$ – инверсия прямо адресуемого бита;
- A – регистр-аккумулятор;
- PC – регистр-счетчик команд;
- $DPTR$ – 16-разрядный регистр-указатель данных;
- $()$ – содержимое ячейки памяти или регистра.

Команды пересылки данных микроконтроллера 8051 включают 28 команд, краткое описание которых приведено в таблице 2.1.

Таблица 2.1 – Команды пересылки данных

<i>Мнемоника</i>	<i>Описание</i>
<i>MOV A, Rn</i>	$(A) \leftarrow (Rn)$
<i>MOVA, addr</i>	$(A) \leftarrow (addr)$
<i>MOV A, @Ri</i>	$(A) \leftarrow ((Ri))$
<i>MOV A, #direct8</i>	$(A) \leftarrow \#direct8$
<i>MOV Rn, A</i>	$(Rn) \leftarrow (A)$
<i>MOV Rn, addr</i>	$(Rn) \leftarrow (addr)$
<i>MOV Rn, #direct8</i>	$(Rn) \leftarrow \#direct8$
<i>MOV add, A</i>	$(addr) \leftarrow (A)$
<i>MOV add, Rn</i>	$(addr) \leftarrow (Rn)$
<i>MOV dst, src</i>	$(dst) \leftarrow (src)$
<i>MOV addr, @Ri</i>	$(addr) \leftarrow ((Ri))$
<i>MOV addr, #direct8</i>	$(addr) \leftarrow \#direct8$
<i>MOV @Ri, A</i>	$((Ri)) \leftarrow (A)$
<i>MOV @Ri, addr</i>	$((Ri)) \leftarrow (addr)$
<i>MOV @Ri, #direct8</i>	$((Ri)) \leftarrow \#direct8$
<i>MOV DPTR, #direct16</i>	$(DPTR) \leftarrow \#direct16$
<i>MOVC A, @A+DPTR</i>	$(A) \leftarrow ((A) + (DPTR))$
<i>MOVC A, @A+PC</i>	$(PC) \leftarrow (PC+1), (A) \leftarrow ((A)+(PC))$

Продолжение таблицы 2.1

<i>MOVX A, @Ri</i>	$(A) \leftarrow ((Ri))$
<i>MOVX A, @DPTR</i>	$(A) \leftarrow ((DPTR))$
<i>MOVX @Ri, A</i>	$((Ri) \leftarrow (A))$
<i>MOVX @DPTR, A</i>	$((DPTR)) \leftarrow (A)$
<i>PUSH addr</i>	$(SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (addr)$
<i>POPaddr</i>	$(addr) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
<i>XCHA.Rn</i>	$(A) \leftrightarrow (Rn)$
<i>XCH A, addr</i>	$(A) \leftrightarrow (addr)$
<i>XCHA, @Ri</i>	$(A) \leftrightarrow ((@Ri))$
<i>XCHDA, @Ri</i>	$(A0 - 3) \leftrightarrow ((@Ri0 - 3))$
<i>MOV A,Rn</i>	$(A) \leftarrow (Rn)$

Команда *MOV* переносит данные из второго операнда в первый. Эта команда не работает с данными, хранящимися во внешней памяти данных или в памяти программ: команда *MOVX* используется для работы с данными, хранящимися во внешней памяти данных, а команда *MOVC* – для работы с константами, хранящимися в памяти программ. Первая из этих команд считывает и записывает байты из внешней памяти данных, а вторая считывает байты из памяти программ.

Команда *XCH* обменивает байты между аккумулятором и ячейкой памяти, а команда *XCHD* обменивает данные между младшими тетрадами (биты от 0 до 3).

Команды *PUSH* и *POP* используются для записи данных в стек и чтения данных из стека соответственно. Размер стека ограничен только размером встроенной памяти данных. При инициализации микроконтроллера после сигнала сброса или подачи напряжения питания в указатель стека *SP* записывается код *07h*. Таким образом, первый элемент стека помещается в ячейку памяти по адресу *08h*.

В группе команд передачи данных микроконтроллера нет команд для работы с регистрами специальных функций (таймеры, порты ввода-вывода и т. д.). Доступ к таким регистрам осуществляется либо непосредственно по адресу, либо с помощью мнемоник, записанных в специальном файле.

Следует отметить, что регистры-аккумуляторы имеют два разных названия в зависимости от способа адресации (*A*– при неявной адресации, например, *MOVA, R0*; *ACC*– при прямой адресации).

Для реализации данного проекта микроконтроллер серии 8051 будет применяться для проектирования системы автоматизированного управления газодинамическим стендом, т. к. удовлетворяет всем необходимым характеристикам.

2.4 Система автоматизированного управления газодинамическим стендом

Структурная схема автоматизированной системы управления газодинамическим стендом (приложение Б) представлена на рисунке 2.4, а схема электрическая принципиальная на рисунке 2.5. Данная система реализована с помощью блочно-иерархического подхода. На вершине данной системы располагается главный модуль системы управления. Он организован по принципу сервера. С помощью ввода информации оператор газодинамического стенда устанавливает параметры температуру нагрева термостата в градусах Цельсия, значение сброса избыточного ПГС в процентах, объем разбавления ПГС воздухом в л/мин и установления влажности разбавляемого воздуха в процентах.

Вывод информации позволяет оператору газодинамического стенда в реальном времени контролировать значения установленных параметров и при необходимости корректировать каждый параметр.

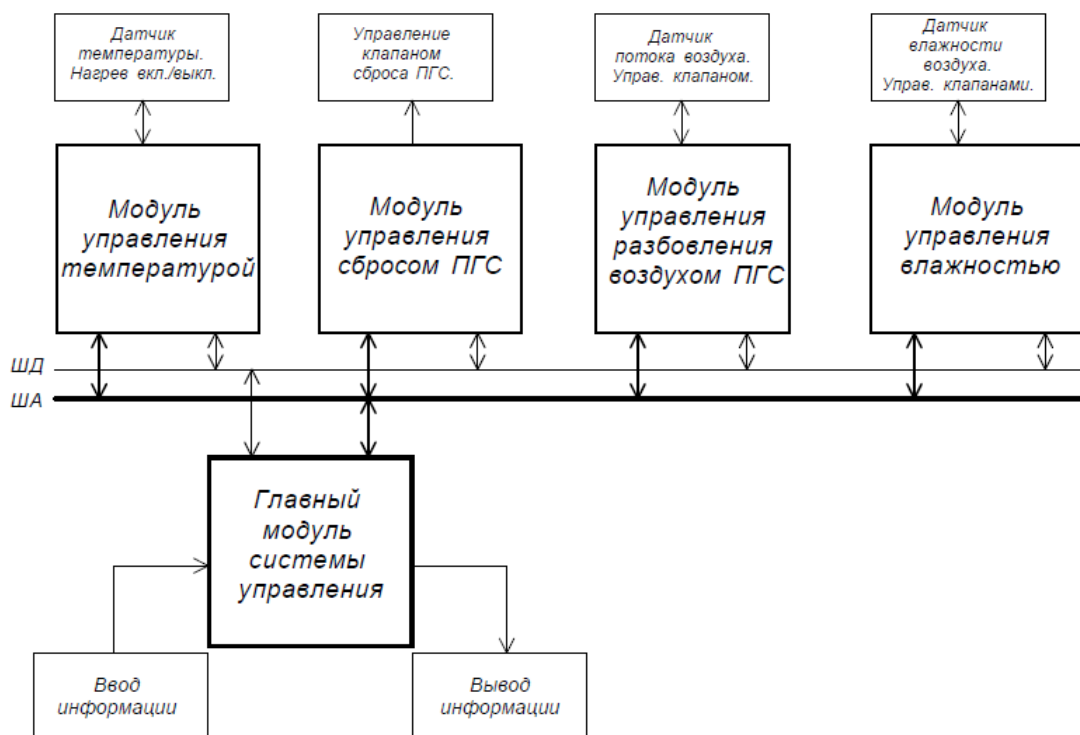


Рисунок 2.4 – Структурная схема системы автоматизированного управления газодинамическим стендом

Чуть ниже в блочно-иерархической системе располагаются модули управления, которые функционируют независимо друг от друга под контролем главного модуля. Модули управления реализованы по принципу клиента. Далее по иерархии располагаются датчики измерения и системы регулирования конкретными термодинамическими параметрами.

Модуль управления температуры функционирует самостоятельно, где производится периодическое измерение температуры в термостате. Когда снижается температура в термостате, нагрев автоматически включается. При достижении установленного значения параметра температуры нагрев – отключается. Этот процесс зафиксирован в бесконечном цикле. При поступлении команды от главного модуля на установку нового значения температуры, модуль в автоматическом режиме переустанавливает свои параметры. При поступлении команды от главного модуля на считывание параметров измерения, модуль в автоматическом режиме передает измеренное значение

в главный модуль управления. После прихода команды на отключение, модуль управления температурой переходит в исходное состояние.

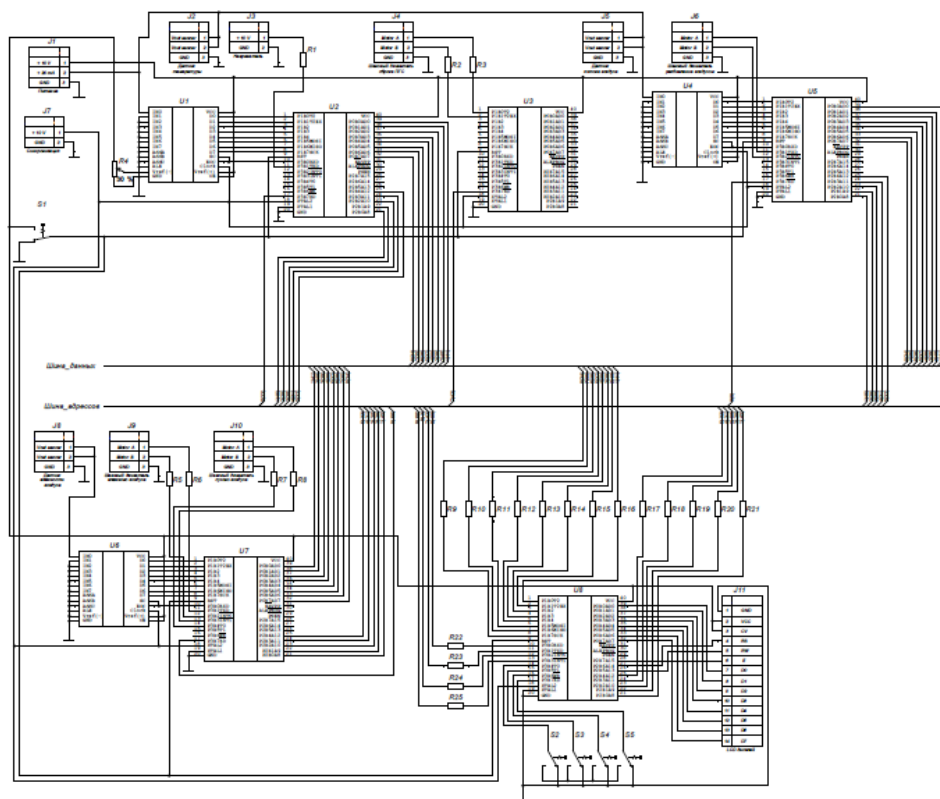


Рисунок 2.5 – Схема электрическая принципиальная системы автоматизированного управления газодинамическим стендом

Модуль управления сбросом ПГС также как модуль управления температурой, работает самостоятельно. Сброс реализован в виде обыкновенного вентиля (клапана), который по команде главного модуля открывает или закрывает трубопровод ПГС на заданный процент позиции штока. При получении команды от главного модуля, происходит автоматическое регулирование положение позиции вентиля клапана, а при завершении работы газодинамического стенда, клапан сброса переходит в исходное положение.

Модуль управления разбавления ПГС воздухом работает самостоятельно под контролем главного модуля. Здесь реализация осуществляется средствами измерения потока воздуха совместно с

регулирующими клапанами. В режиме реального времени модуль управления разбавлением устанавливает и измеряет значение потока подаваемого воздуха в ПГС. При завершении работы газодинамического стенда, модуль управления разбавлением переходит в исходное состояние.

Модуль управления влажностью по аналогии с вышеперечисленными модулями, работает самостоятельно под контролем главного модуля. Этот модуль управляет влажностью разбавляемого воздуха средствами измерения и регулирования. К средствам измерения относится датчик влажности, а к средствам регулирования два клапана влажного и сухого воздуха, которые открывают или закрывают трубопроводы в газодинамическом стенде.

Для проектирования системы автоматизированного управления газодинамическим стендом воспользуемся принципом декомпозиции. Декомпозиция – разбиение схемы на части. Выполним проектирование каждого модуля по отдельности, а потом объединим их в одну систему.

2.5 Разработка модуля управления температурой

Для реализации модуля управления температурой необходимо использовать микроконтроллер(MCU 8051), аналого-цифровой преобразователь (АЦП)(ADC 0808), датчик температуры (Pt 100).

Определим диапазон необходимого напряжения для контроля температуры в диапазоне от 20 до 99 °С. Рассмотрим характеристики датчика температуры Pt 100 [10]. На рисунке 2.6 представлена зависимость температуры от электрического сопротивления чувствительного элемента.

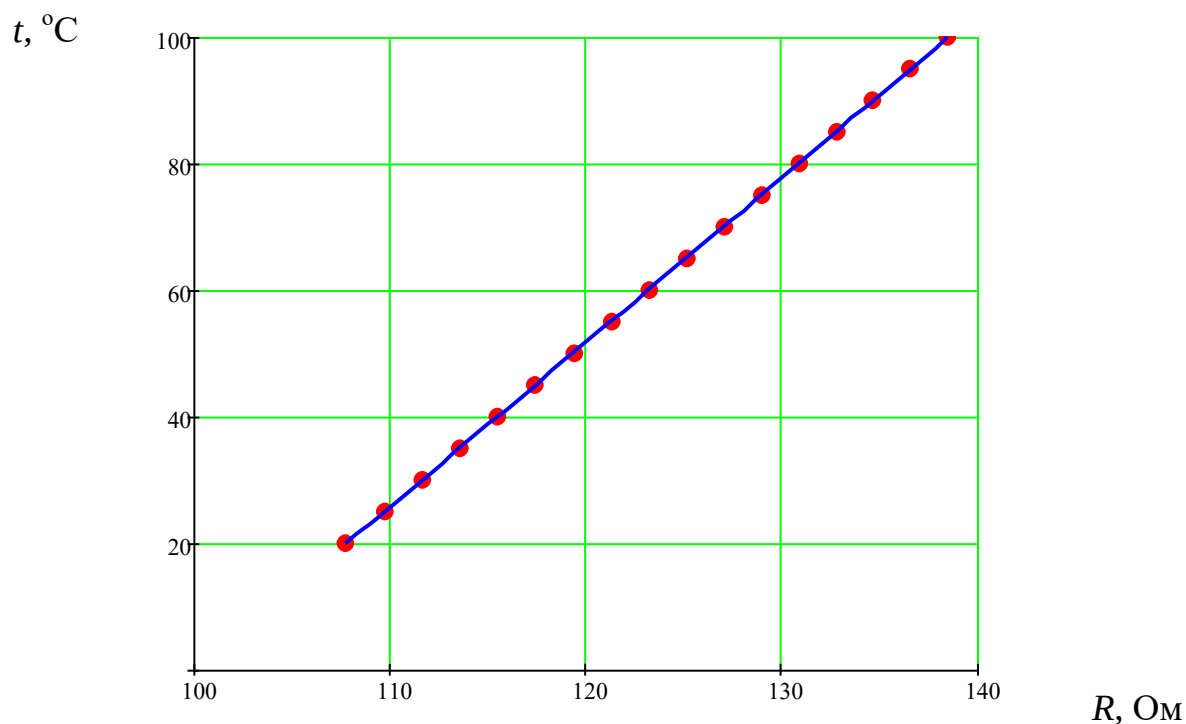


Рисунок 2.6 – Зависимость температуры от электрического сопротивления

На основании зависимости температуры от сопротивления с помощью линейной регрессии найдем математическую зависимость, которая имеет вид:

$$t(R) = 2,60 \cdot R - 260,35 \text{ (}^\circ\text{C)}, \quad (2.1)$$

Рассчитаем необходимое напряжение при постоянном токе $I = 20 \text{ мА}$:

$$U_{\min} = I \cdot R = 20 \times 10^{-3} \cdot 107,79 = 2,16 \text{ (В)}, \quad (2.2)$$

$$U_{\max} = I \cdot R = 20 \times 10^{-3} \cdot 138,51 = 2,77 \text{ (В)}. \quad (2.3)$$

Вычислим разницу напряжений ΔU между максимальным и минимальным значениями напряжения:

$$\Delta U = U_{\max} - U_{\min} = 2,77 - 2,16 = 0,61 \text{ (В)}. \quad (2.4)$$

Рассчитаем необходимый коэффициент усиления наиболее приближающего к 10 В, напряжения питания микроконтроллера $U_{\text{пит}}$:

$$k_y = \frac{U_{\text{пит}}}{U_{\text{max}}} = \frac{10}{2,77} \approx 3,61. \quad (2.5)$$

Рассчитаем входное опорное напряжения $U_{\text{ref}(+/-)}$ для АЦП, чтобы обеспечить точность измерения:

$$U_{\text{ref}(+)} = U_{\text{max}} \cdot k_y = 2,77 \cdot 3,6 \approx 9,99 \approx 10 \text{ (В)}, \quad (2.6)$$

$$U_{\text{ref}(-)} = U_{\text{min}} \cdot k_y = 2,16 \cdot 3,6 \approx 7,78 \approx 8 \text{ (В)}. \quad (2.7)$$

Т. к. порты ввода / вывода ($P0-P3$) микроконтроллер 8051 обеспечивает 1 байтинформации, рассчитаем коэффициент преобразования напряжения в шестнадцатеричную систему счисления:

$$k_{\text{hex}} = \frac{256}{U_{\text{ref}(+)} - U_{\text{ref}(-)}} = \frac{256}{10 - 8} = 128. \quad (2.8)$$

Коэффициент обратного преобразования из шестнадцатеричной системы в значение напряжения на датчике с учетом коэффициента усиления будет составлять: $1 / 128 \approx 0,008$.

Полученные коэффициенты необходимы для программы микроконтроллера, которая управляет работой модуля управления температурой.

На рисунке 2.7 представлена модель работы модуля управления температурой.

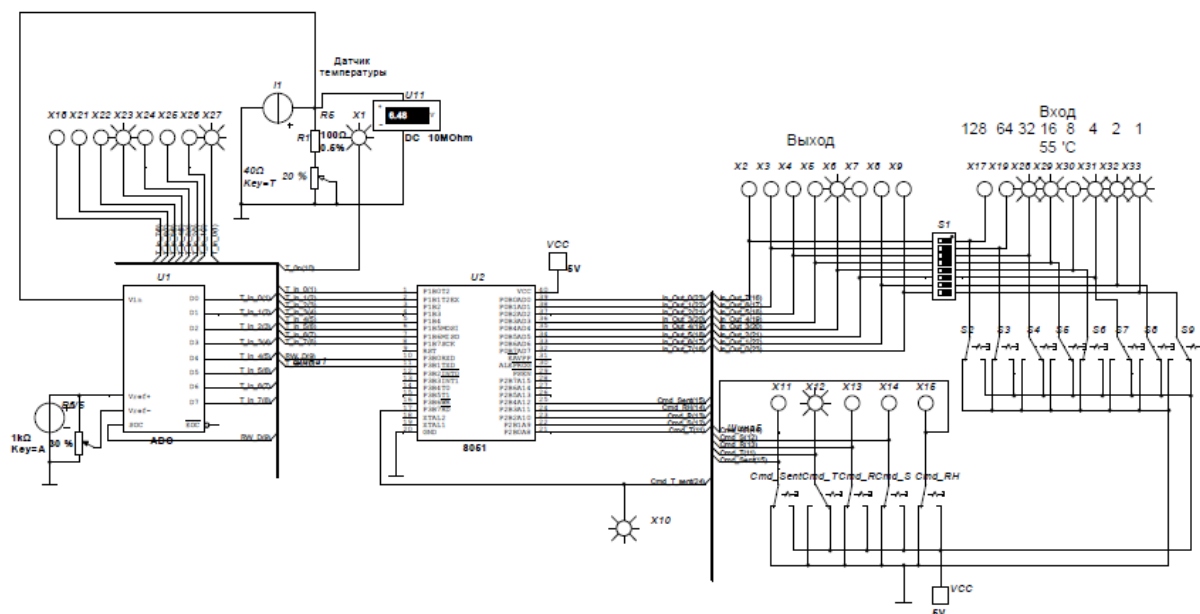


Рисунок 2.7 – Модель работы модуля управления температурой

После включения работы модуля управления температурой микроконтроллер в соответствии с программой (Приложение В) переходит в рабочее состояние, оповестив главный модуль сигналом $X10$ (рисунок 2.7). После чего оператор газодинамического стенда на главном модуле установил значение температуры, например 55°C .

С помощью моделирования сигналов на вход микроконтроллера приходит команда установить 55°C (в двоичной системе счисления 00110111) для термостата газодинамического стенда. Микроконтроллер записывает данный параметр в память и включает нагрев термостата, что показано сигналом $X1$ (рисунок 2.7). На рисунке 2.7 видно как аналоговый сигнал представлен в виде шестнадцатеричного кода, который преобразует АЦП. Данный код сравнивается в программе микроконтроллера с установленным значением температуры и при значении меньше установленного включается нагрев термостата, а при большем значении нагрев отключается. Главный модуль управления посылает команду на считывания измеренной температуры (светодиод $X12$ горит). Микроконтроллер преобразует шестнадцатеричный код измеренной

температуры в двоичный код, и посылает на главный модуль управления, где отображается информация уже в десятичной системе, понятной для оператора газодинамического стенда. На выходе микроконтроллера в двоичном коде имеет значение 00001000, что соответствует в десятичной форме 8 °С. При этом микроконтроллер посылает команду на подтверждения принятой информации (светодиод Х10 – горит).

Таким образом, работа модуля управления температурой работает в соответствие с заданным алгоритмом, представленным в приложении В.

2.6 Разработка модуля управления сбросом парогазовой смеси

Для реализации модуля управления сбросом парогазовой смеси необходимо использовать микроконтроллер(MCU 8051), шаговый двигательFL28STH32-0956A [11], который своим валом с помощью соединительной муфты подключается к вращающему вентилю расхода воздуха (рисунок 2.8) трубопровода [12].



Рисунок 2.8 – Регулятор расхода воздуха

Рассчитаем необходимые коэффициенты для реализации программы микроконтроллера. Расчет проведем по имеющимся характеристикам регулятора расхода воздуха и шагового двигателя.

Для полного закрытия вентиля расхода воздуха необходимо совершить полтора оборота, что составляет $360^{\circ} + 180^{\circ} = 540^{\circ}$. Один импульс шаговый

двигатель поворачивает вал на $1,8^\circ$. Вычислим количество импульсов необходимых для полного закрытия вентиля регулятора воздуха с помощью шагового двигателя:

$$n_{\text{имп}} = \frac{540}{1,8} = 300 \quad (\text{импульса}). \quad (2.9)$$

Рассчитаем коэффициент преобразования количества импульсов в проценты, т.к. контроль сброса ПГС контролируется в процентах от закрытия вентиля:

$$k_{\text{имп}_\%} = \frac{99}{300} = 0,33. \quad (2.10)$$

Коэффициент обратного преобразования процентов в импульсы будет составлять: $1 / 0,33 \approx 3,03$.

На рисунке 2.9 представлена модель работы модуля управления сбросом ПГС.

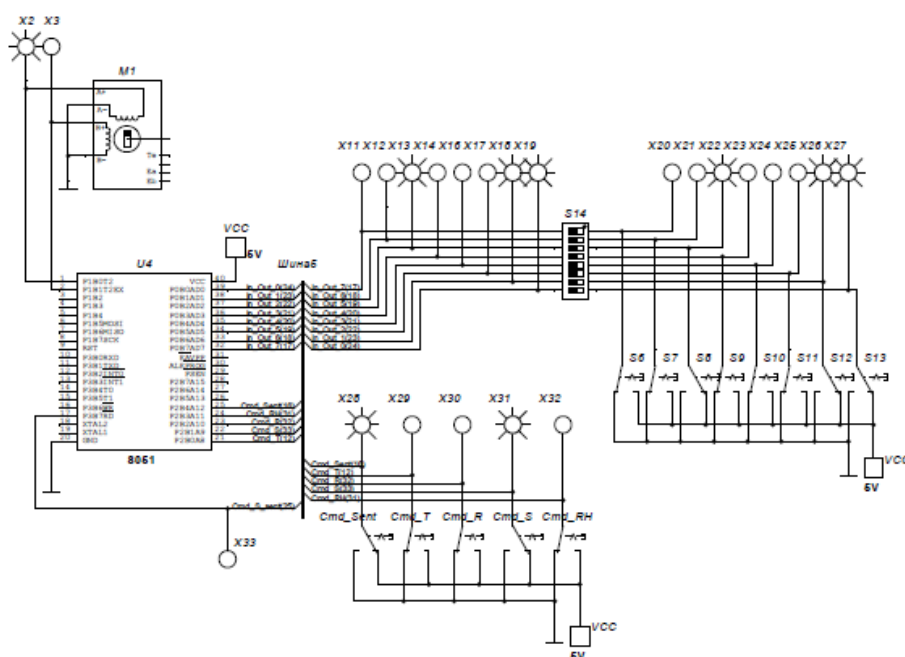


Рисунок 2.9 – Модель работы модуля управления сбросом

После включения работы модуля управления сбросом ПГС микроконтроллер в соответствии с программой (Приложение В) инициализирует свою работу. После инициализации микроконтроллер посылает команду главному модулю о готовности. Ввиду того, что механизм сброса излишней ПГС реализован в виде простого регулятора расхода воздуха подключенного к шаговому двигателю *M1* (рисунок 2.9), то микроконтроллер просто ожидает команды от главного модуля на сколько необходимо закрыть вентиль в процентном отношении. В данном случае моделируется значение в виде двоичного кода 00100011 (рисунок 2.9), что в десятичном варианте означает 35 %. После установления данного значения главный модуль посылает команду установить это значение (светодиоды *X28* и *X31* – горят) (рисунок 2.9). После получение команды микроконтроллер посылает импульсы на обмотки шагового двигателя *M1* (светодиод *X2* – горит). После расчетного количества импульсов микроконтроллер с помощью шагового двигателя закроет регулятор потока воздуха до уровня 35 %, а потом отправит команду (светодиод *X33* загорится) главному модулю управления, что готов к следующей операции. В режиме реального времени главный модуль будет получать эти показания и сигнализировать оператору газодинамического стенда о состоянии сброса избыточной ПГС.

Таким образом, работа модуля управления сбросом избыточной ПГС работает в соответствие с заданным алгоритмом, представленным в приложении В.

2.7 Разработка модуля управления разбавления воздухом

Для реализации модуля управления разбавления воздухом ПГС необходимо использовать микроконтроллер(*MCU 8051*), АЦП(*ADC 0808*), датчик потока воздуха (*EE650*), шаговый двигатель(*FL28STH32-0956A*), который своим валом с помощью соединительной муфты подключатся к вращающему вентилю расхода воздуха (рисунок 2.8) трубопровода.

Рассчитаем диапазон напряжения для контроля потока воздуха в диапазоне от 0 до 30 л/мин. Рассмотрим характеристики датчика потока воздуха EE650 [13]. На рисунке 2.10 представлена зависимость скорости потока воздуха от напряжения.

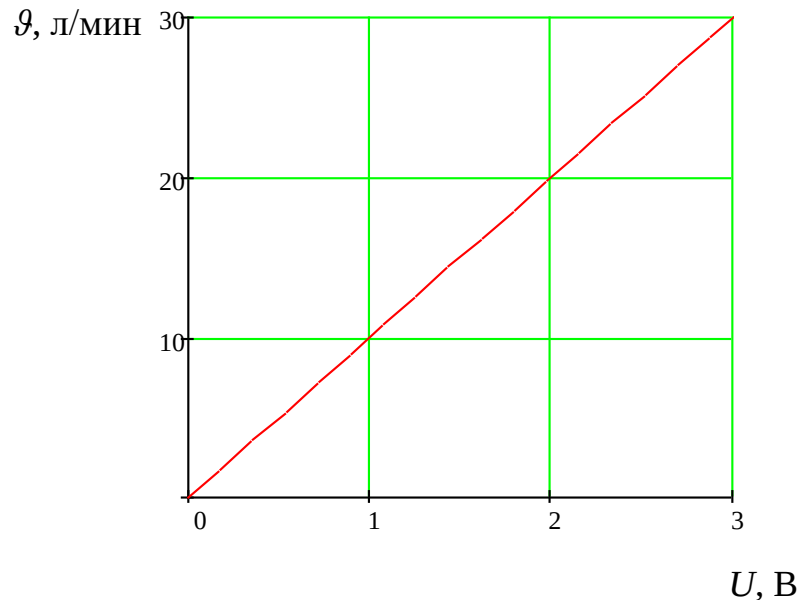


Рисунок 2.10 – Зависимость скорости потока воздуха от напряжения

На основании зависимости потока воздуха от напряжения с помощью линейной регрессии найдем математическую зависимость, которая имеет вид:

$$Q(U) = 10 \cdot U \quad (\text{л/мин}). \quad (2.11)$$

Из анализа рисунка 2.10 можно сделать выводы, что минимальное и максимальное напряжения равны $U_{min} = 0$ и $U_{max} = 3$ В. Разница напряжения $\Delta U = 3$ В. Следовательно входное опорное напряжение для АЦП составляет $U_{ref(-)} = 0$ В, $U_{ref(+)} = 3$ В.

Т. к. порты ввода / вывода (P0-P3) микроконтроллер 8051 обеспечивает 1 байт информации, рассчитаем коэффициент преобразования напряжения в шестнадцатеричную систему счисления:

$$k_{hex} = \frac{256}{U_{ref(+)} - U_{ref(-)}} = \frac{256}{3} \approx 85,33. \quad (2.8)$$

Коэффициент обратного преобразования из шестнадцатеричной системы в значение напряжения на датчике составлять: $1 / 85,33 \approx 0,012$.

Полученные коэффициенты необходимы для программы микроконтроллера (Приложение В), которая управляет работой модуля управления разбавления воздухом ПГС.

Расчет параметров шагового двигателя было представлено выше.

На рисунке 2.11 представлена модель работы модуля управления разбавления воздухом ПГС.

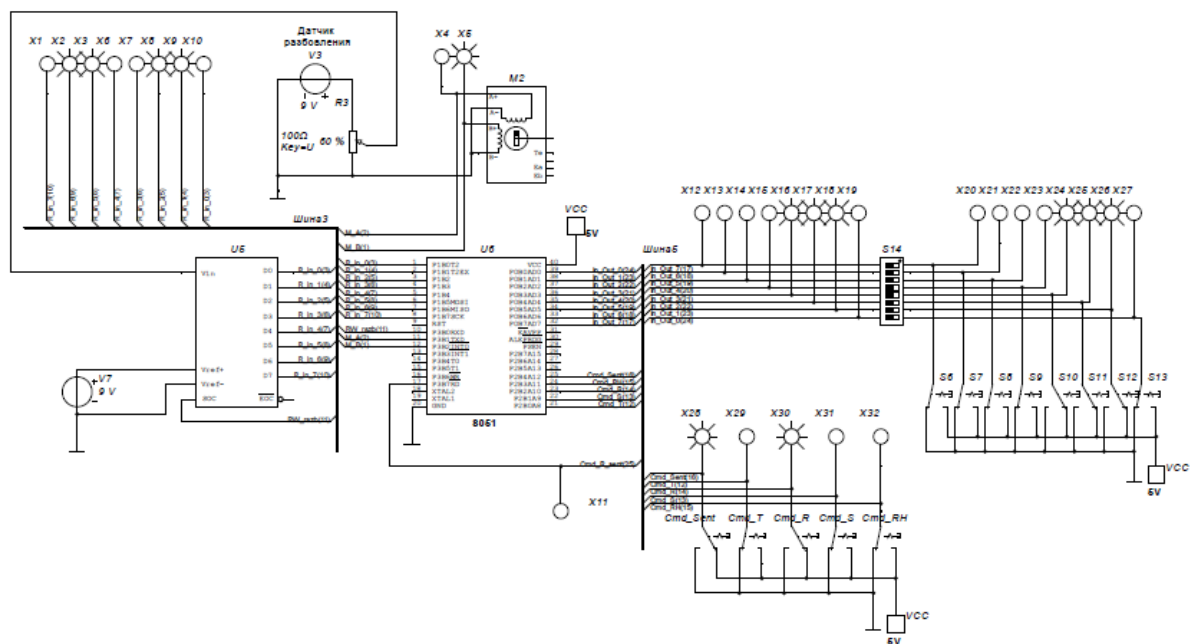


Рисунок 2.11 – Модель работы модуля управления разбавления воздухом ПГС

После включения ГДС модуль управления разбавления воздухом ПГС в соответствии с программой (Приложение В) выполняет инициализацию. Микроконтроллер посылает команду о готовности главному модулю управления ГДС и ожидает ответную команду.

Здесь на рисунке 2.11 представлен этап установления параметра разбавления воздухом от главного модуля. Главный модуль управления ГДС посылает установленное значение по шине данных в виде двоичного кода 00001110, что соответствует 14 л/мин, а по шине адресов посылается команда на установку параметра для модуля управления разбавлением воздуха ПГС (светодиоды X28 и X30 горят). Как видно на этом рисунке выполняется работа шагового двигателя по открытию вентиля в заданное положение. При этом в режиме реального времени датчик потока воздуха выполняет измерение. После того, как шаговый двигатель закончит установку вентиля, микроконтроллер посылает команду о завершении данной операции.

Таким образом, работа модуля управления воздухом ПГС работает в соответствии с заданным алгоритмом, представленным в приложении В.3.

2.8 Разработка модуля управления влажностью воздуха

Для реализации модуля управления влажностью воздуха необходимо использовать микроконтроллер(MCU 8051), АЦП(ADC 0808), датчик влажности воздуха (НТР 201), шаговый двигатель(FL28STH32-0956A), который своим валом с помощью соединительной муфты подключается к вращающему вентилю расхода воздуха (рисунок 2.8) влажного и сухого трубопровода.

Рассчитаем диапазон напряжения для контроля влажности воздуха в диапазоне от 0 до 90 %. Рассмотрим характеристики датчика влажности воздуха НТР 201 [14]. На рисунке 2.12 представлена зависимость влажности воздуха от напряжения.

На основании зависимости влажности воздуха от напряжения с помощью линейной регрессии найдем математическую зависимость, которая имеет вид:

$$RH(U)=10 \cdot U \quad (\%). \quad (2.11)$$

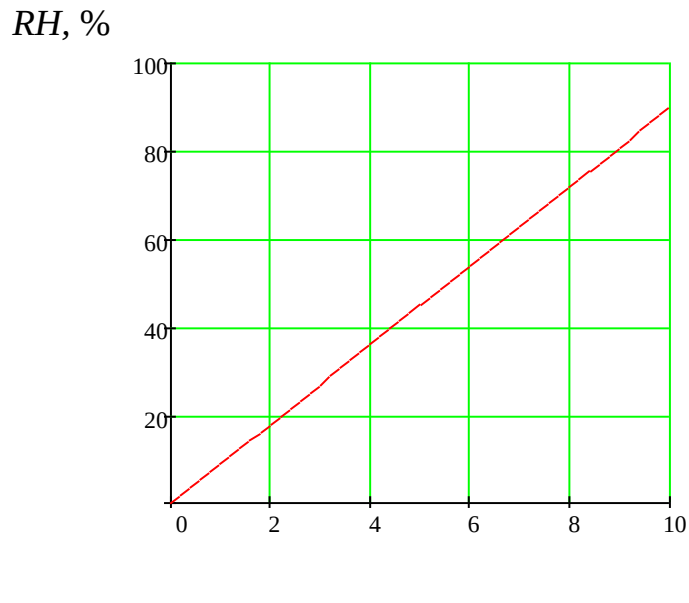


Рисунок 2.12 – Зависимость влажности воздуха от напряжения

Из анализа рисунка 2.12 можно сделать выводы, что минимальное и максимальное напряжения равны $U_{min} = 0$ и $U_{max} = 10$ В соответственно. Разница напряжения $\Delta U = 10$ В. Следовательно входное опорное напряжение для АЦП составляет $U_{ref(-)} = 0$ В, $U_{ref(+)} = 10$ В.

Т. к. порты ввода / вывода (P0-P3) микроконтроллер 8051 обеспечивает 1 байт информации, рассчитаем коэффициент преобразования напряжения в шестнадцатеричную систему счисления:

$$k_{hex} = \frac{256}{U_{ref(+)} - U_{ref(-)}} = \frac{256}{10} = 25,6. \quad (2.8)$$

Коэффициент обратного преобразования из шестнадцатеричной системы в значение напряжения на датчике составлять: $1 / 25,6 \approx 0,039$.

Расчет параметров шагового двигателя было представлено выше.

Полученные коэффициенты необходимы для программы микроконтроллера (Приложение В), которая управляет работой модуля управления влажностью воздуха.

На рисунке 2.13 представлена модель работы модуля управления влажностью воздуха.

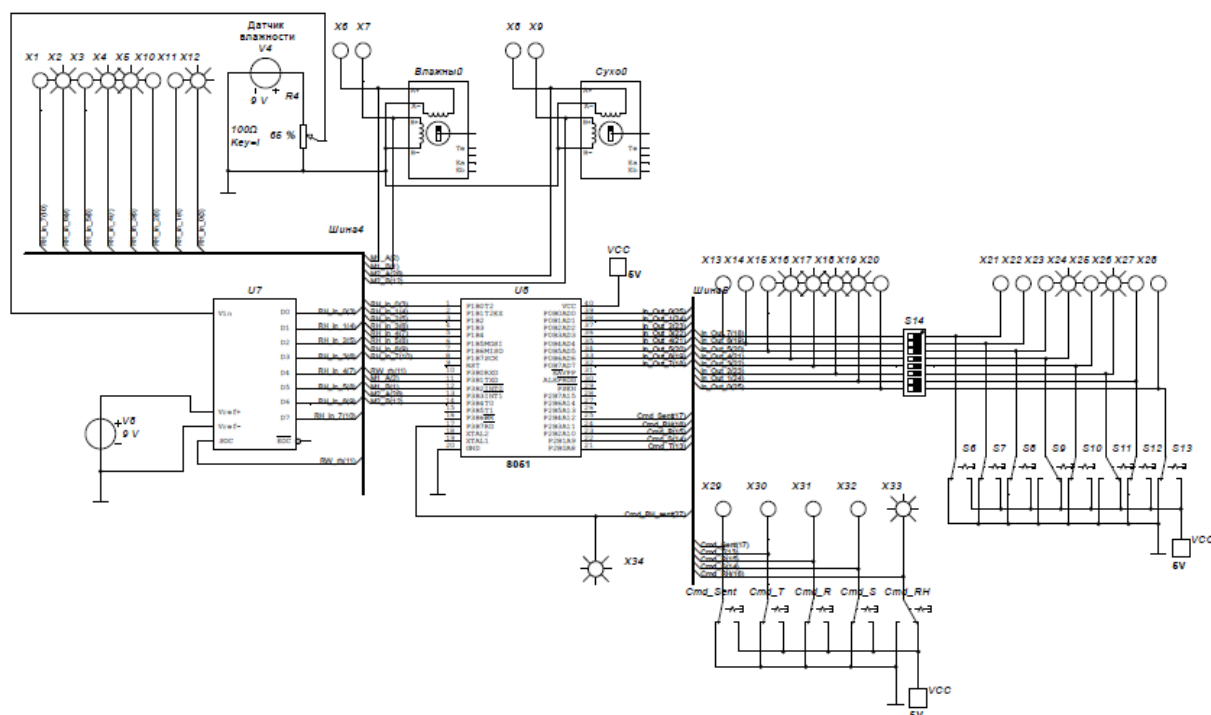


Рисунок 2.13 – Модель работы модуля управления влажностью воздуха

При включении ГДС модуль управления влажностью воздуха с помощью программы микропроцессора инициализирует работу. Шаговые двигатели поворачивают вентили влажного и сухого расхода воздуха на половину вентилля. Таким образом, создается примерная влажность воздуха около 50 %. После того, как шаговые двигатели переведут вентили (сухой и влажный) трубопроводы в положение 50 %, модуль управления влажностью воздуха передает команду главному модулю управление на готовность. В процессе всего процесса эксплуатации ГДС, в режиме реального времени идет контроль влажности воздуха для разбавления ПГС.

На рисунке 2.13 представлен этап передачи измеренной влажности воздуха в главный модуль. Главный модуль управления ГДС подает команду на измерения влажности воздуха (светодиод X33 горит). После получения

данной команды модуль управления влажностью воздухом по шине данных передает полученное измерение в виде двоичного кода 00011110, что в десятичной системе означает 30 %. Как только модуль управления влажностью отправил измеренные данные главному модулю, посылается команда о завершении передачи измерения параметра (светодиод ХЗ4 – горит). При этом модуль управления влажностью в автоматическом режиме проводит подстройку влажного и сухого трубопровода в установленном диапазоне, приблизительно 20 % от установленного значения главным модулем.

Таким образом, модуль управления влажностью работает в соответствии с алгоритмом, представленным в приложении В.

2.9 Разработка главного модуля управления

Для реализации главного модуля управления ГДС необходимо использовать микроконтроллер (*MCU 8051*), *LCD*-дисплей (*MT-08S2A*) [15], четыре микропереключателей для установки параметров работы ГДС (*CHEERBRIGHT*, *DC12V*, *0.5A*, *3×4×3 мм*, *3 Pin*).

При включении ГДС главный модуль управления выполняет инициализацию своей работы в режиме реального времени в соответствии с алгоритмом программы (Приложение В). После инициализации главный модуль управления выполняет опрос командой о готовности других модулей к работе. На рисунке 2.14 представлен фрагмент моделирования инициализации работы главного модуля и опрос других модулей (светодиод *Cm_Sent* горит). *LCD*-дисплей сигнализирует оператору «*GDSready*», что означает готовность работы ГДС и готовность установки термодинамических параметров для ПГС. Термодинамическими параметрами ГДС является температура нагревания источника ПГС в °С, сброс излишней ПГС контрольного продукта в %, скорость разбавления ПГС воздухом в л/мин и влажность этого воздуха в %.

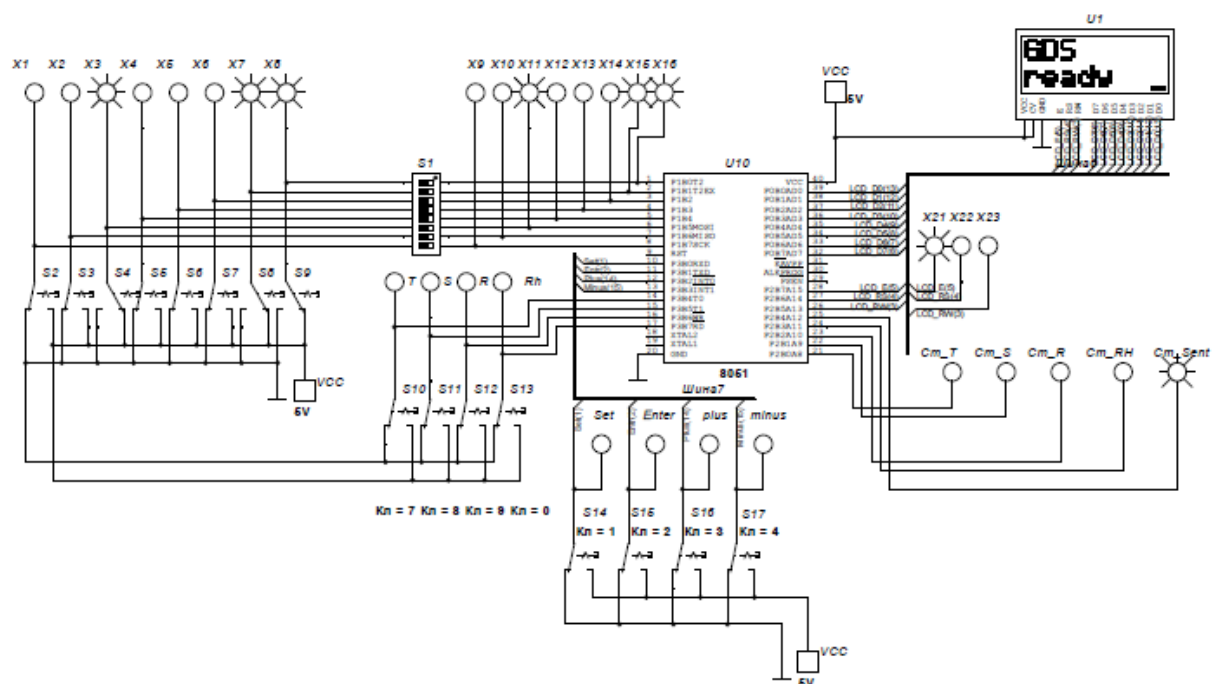


Рисунок 2.14 – Фрагмент моделирования работы главного модуля управления ГДС состояние «готовность»

Для установления термодинамических параметров используются четыре микропереключателей выполняющие следующие функции:

- микропереключатель S14, светодиод *Set* предназначен для выбора термодинамического параметра ПГС и выхода из режима установки;
- микропереключатель S15, светодиод *Enter* предназначен для ввода установленного параметра ПГС;
- микропереключатель S16, светодиод *Plus* предназначен для увеличения значения параметра ПГС;
- микропереключатель S17, светодиод *Minus* предназначен для уменьшения значения параметра ПГС.

На рисунке 2.15 представлен фрагмент моделирования работы главного модуля управления ГДС в режиме установки и ввода параметра температуры.

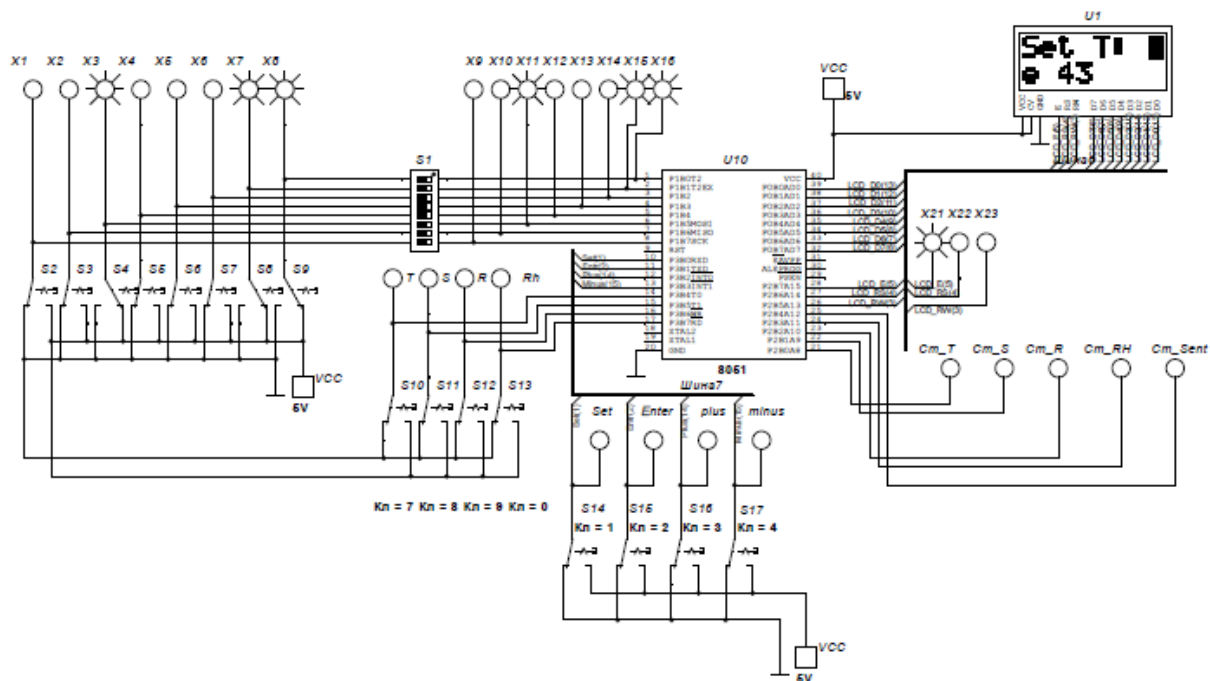


Рисунок 2.15 – Фрагмент моделирования работы главного модуля управления ГДС состояние «установка температуры»

Для входа в режим установки термодинамических параметров оператор нажимает кнопку микропереключателя *Set*. Главный модуль управления переходит в режим «установка». С помощью микропереключателей *plus* и *minus* оператор может переключать разные параметры в циклическом режиме. После выбора нужного параметра с помощью микропереключателя *enter* входит в подменю для увеличения или уменьшения значения параметра. С помощью микропереключателей *plus* и *minus* оператор увеличивает или уменьшает значение термодинамического параметра соответственно. Потом оператор может нажать микропереключатель *enter* для записи данного параметра и последующего отправления значения по шине данных на соответствующий модуль управления термодинамическим параметром, в противном случае он может нажать микропереключатель *set* для отмены установленного параметра и выхода в меню «установка».

После установки всех необходимых термодинамических параметров оператор для выхода из меню «установка» нажимает микропереключатель *set* и главный модуль управления ГДС переходит в режим измерения

параметров. На рисунке 2.16 представлен фрагмент моделирования измерения термодинамических параметров.

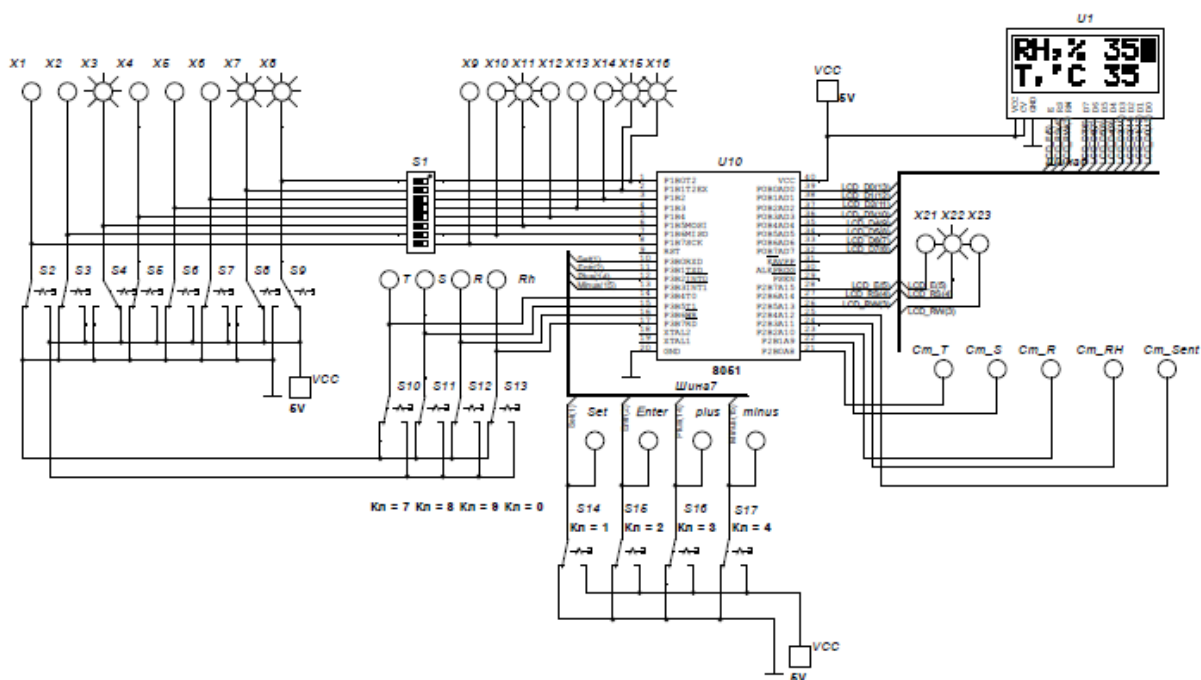


Рисунок 2.16 – Фрагмент моделирования работы главного модуля управления ГДС состояние «измерения»

Измерения работает в режиме реального времени. Главный модуль управления ГДС в бесконечном цикле посылает по очереди команды «измерения» каждому модулю, что показано светодиодами на рисунке 2.16: *Cm_T* – команда «измерения» модулю управления температурой, *Cm_S* – команда «измерения» модулю управления сбросом избыточной ПГС, *Cm_R* – команда «измерения» модулю управления разбавлением воздухом ПГС, *Cm_RH* – команда «измерения» модулю управления влажностью воздуха. После получения команды «измерения» каждый модуль по шине данных посылает измеренное значение главному модулю управления ГДС. На рисунке 2.16 показано в двоичном коде измеренное значение 00100011, что соответствует в десятичном виде 35. После передачи информации модуль конкретного параметра посылает команду о завершении передачи информации об измеренном значении в виде следующих сигналов,

что показано светодиодами на рисунке 2.16: T – завершение передачи измеренной информации от модуля управления температурой, S – завершение передачи о состоянии открытия вентиля сброса от модуля управления сбросом избыточной ПГС, R – завершение передачи измеренной информации от модуля управления разбавлением и Rh – завершение передачи измеренной информации от модуля управления влажностью воздуха. Как видно на рисунке 2.16 измеренная информация отображается на LCD -дисплее.

По завершению работы ГДС оператор одновременно нажимает и удерживает не менее 5 секунд два микропереключателя *plus* и *minus* для завершения работы. Главный модуль управления переходит в завершающий режим и посылает команду «завершить работу» модулям управления термодинамическими параметрами ПГС. После получения такой команды каждый модуль завершает свою работу и переводит управляющие механизмы в первоначальное состояние. На LCD -дисплее появляется информация о завершении работы ГДС, что показано на рисунке 2.17.

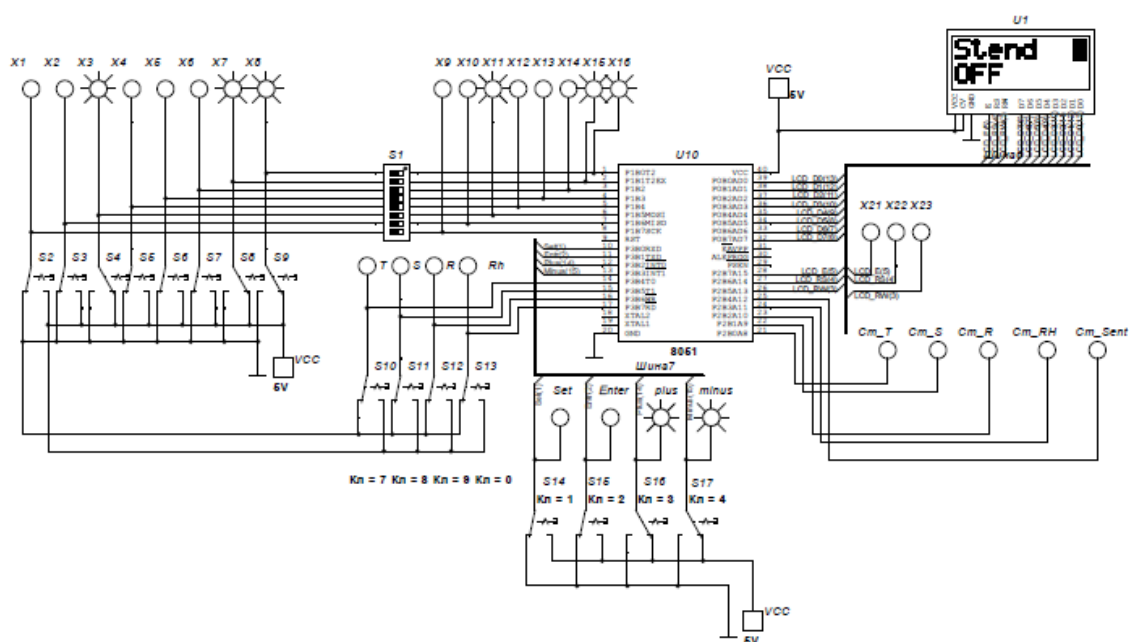


Рисунок 2.16 – Фрагмент моделирования работы главного модуля управления ГДС состояние «завершение»

Таким образом, главный модуль управления работает в соответствии с алгоритмом, представленным в приложении В.

2.10 Автоматизированная работа газодинамического стенда

В результате разработки системы автоматизированного управления ГДС с помощью программируемых микропроцессоров осуществляется стабильная работа термодинамической системы в ГДС. Оператор стенда принимает минимальное участие в процесс создания парогазовой смеси контрольного образца, что позволяет ему все свое время посвятить исследованию чувствительности газовых сенсоров. Как видно на рисунке 2.17 стенд состоит из отдельных модулей соединенных лишь трубопроводами, а система автоматизированного управления ГДС позволяет в комплексе и с высокой точностью поддерживать термодинамические параметры ПГС, что является ключевым в исследовании чувствительных элементов.

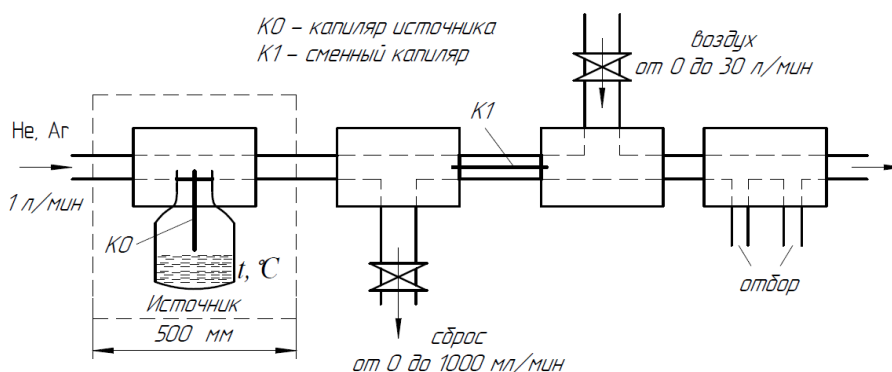


Рисунок 2.17 – Газодинамический стенд

Причем каждый модуль выполняет независимо друг от друга свою функцию в режиме реального времени, а оператор может лишь корректировать установленные параметры в процессе исследования газочувствительности сенсоров.

3 Техника безопасности при эксплуатации электрооборудования

Тема промышленной безопасности на предприятиях – одна из самых актуальных в современной производственной среде. Тенденция к внедрению новых технологий в опасное производственное оборудование и постоянное устаревание промышленных установок обусловили необходимость поиска новых подходов к организации промышленной безопасности [16].

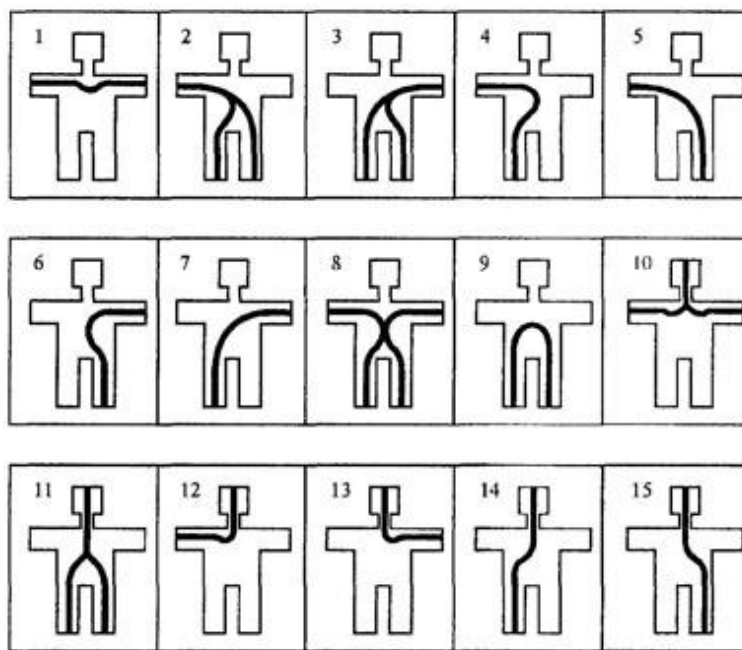
Под электробезопасностью понимаются организационные и технические мероприятия и средства защиты людей от вредного и опасного воздействия электрических токов, излучений, электромагнитных полей и статического электричества. Несоблюдение требований электробезопасности при эксплуатации производственного или электрического оборудования, а также при непосредственном контакте с рабочими органами работающего оборудования может привести к поражению электрическим током. Обеспечение безопасного обслуживания электрооборудования во многом зависит от факторов окружающей среды. Исходя из этих факторов и их наличия, все помещения можно разделить на три категории:

- 1) помещения, не содержащие опасностей;
- 2) помещения с повышенной опасностью; и
- 3) помещения, представляющие особую опасность.

В основном условия, повышающие или понижающие риск поражения электрическим током, характеризуются температурой в помещении, влажностью, наличием токопроводящих поверхностей, величиной химически активной среды и возможностью одновременного контакта металлического корпуса (открытых токопроводящих частей) электрооборудования с человеком.

Согласно исследованиям, большинство несчастных случаев происходит из-за недопустимости работы с электрооборудованием неквалифицированными специалистами и игнорирования средств защиты.

При прохождении электрического тока через организм возникают тепловые, электролитические и биологические воздействия. Тепловое воздействие приводит к ожогам, нагреванию крови и сосудов, электролитическое – к разложению крови, а биологическое – к стимуляции работы тканей, что может привести к прекращению кровообращения и дыхания. Последствия от прохождения тока зависят от его характеристик, продолжительности воздействия, маршрута и общего состояния человека. Даже при одинаковом значении тока, ущерб для организма может быть различным в зависимости от его траектории («петли тока»). На рисунке 3.1 показаны типичные маршруты прохождения тока через человеческое тело.



1 — рука-рука; 2 — правая рука-ноги; 3 — левая рука-ноги; 4 — правая рука-правая нога; 5 — правая рука-левая нога; 6 — левая рука-левая нога; 7 — левая рука-правая нога; 8 — обе руки-обе ноги; 9 — нога-нога; 10 — голова-руки; 11 — голова-ноги; 12 — голова-правая рука; 13 — голова-левая рука; 14 — голова-правая нога; 15 — голова-левая нога

Рисунок 3.1 – Характерные пути тока в человеке

При работе с электрооборудованием необходимо использовать общие и индивидуальные средства защиты, чтобы избежать поражения электрическим током. Основные меры защиты включают:

- 1) заземление и отсоединение корпуса электрооборудования, которое может находиться под напряжением;
- 2) электрическая изоляция частей, находящихся под напряжением;
- 3) защитные кожухи.
- 4) применение безопасного низкого напряжения
- 5) предупреждающие знаки
- 6) знаки безопасности
- 7) использование средств индивидуальной защиты.

Из вышесказанного следует, что производственные помещения и научные лаборатории должны соответствовать требованиям СНиП и санитарно-гигиеническим нормам. Кроме того, для обеспечения безопасности при использовании электроэнергии необходимо реализовать следующие существующие меры:

- 1) соответствующий отбор персонала, обслуживающего электрооборудование;
- 2) контроль технического состояния оборудования в соответствии с ГОСТ Р 56397-2015;
- 3) проведение обучения и инструктажа по правилам техники безопасности;
- 4) назначение лица, ответственного за эксплуатацию электрооборудования.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы было выполнено следующее:

- проведен анализ компонентов электроники используемых для проектов автоматизации, как технологических процессов, так и систем автоматики;
- проведен анализ измерительных приборов применяемых в системе автоматизации;
- проведен анализ исполнительных компонентов применяемых в качестве движущей силы в системах автоматизации;
- проведен анализ микропроцессорной техники, которые упрощают систему управления технологическим процессом;
- разработана автоматизированная система управления газодинамическим стендом, что позволяет упростить процесс создания парогазовых смесей с заданной концентрацией контрольного вещества.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Свидетельство об утверждении типа средств измерений. RU.C.31.004. А№49893 /Установка динамическая «Микрогаз-Ф»;
2. Пат. 200896 РФ. Стенд для газодинамических испытаний/ Авторы: Мазин К.Е., Дымнич С.А., Бойко А.Ю., Жохов А.К, Елизаров А.В., Брагинец А.А., Орлов Е.Д., Костюнин А.П. // Бюл. – 2020. – №32. – С.11.
3. Пат. 92535 РФ. Газодинамическая установка/ Авторы: Мандыч В.Г., Шехтер Г.А., Левшов И.А. идр. // Бюл. – 2010. – №1. – С. 8.
4. Коротун В. Как работает шаговый электродвигатель // ASUTTP URL: <https://www.asutpp.ru/shagovyy-dvigatel.html> (дата обращения: 16.04.2024).
5. Макаров Ю. Что такое терморезисторы, их конструкция, виды, технические параметры // ООО «РадиоЭлемент» URL: <https://yandex.ru/turbo/asutpp.ru/s/termorezistor.html> (дата обращения: 16.04.2024).
6. Газодинамические датчики потока. Датчики потока использующие эффект переноса тепла // URL: <https://meduniver.com/Medical/Xirurgia/2146.html?ysclid=ltkh87k3gz935821888> MedUniver (дата обращения: 17.04.2024).
7. Датчики влажности – как устроены и работают // URL: <http://elektrik.info/main/automation/1083-datchiki-vlazhnosti-kak-ustroeny-i-rabotayut.html>(дата обращения: 20.04.2024).
8. Благовещенская М.М Информационные технологии систем управления технологическими процессами. Учебник для вузов/М.М. Благовещенская, Л.А. Злобин. – М.: Высшая школа, 2005. –768с.
9. Магда Ю.С. Микроконтроллеры серии 8051: практический подход. — М.: ДМК Пресс, 2008. – 228 с.

10. Бескорпусной датчик температуры Pt100 // URL: <https://www.iecon.ru/catalog/datchiki-temperature/beskorpusnye/Pt100/> (дата обращения: 16.04.2024).
11. FL28STH32-0956A Datasheet High Torque Hybrid Stepping Motor // URL: <https://datasheet4u.com/datasheet-pdf/MOTIONCONTROLPRODUCTS/FL28STH32-0956A/pdf.php?id=1079579> (дата обращения: 16.04.2024).
12. Регулятор расхода воздуха SCF6 NBPT 52884 // URL: <https://www.vseinstrumenti.ru/product/regulyator-rashoda-vozdruha-scf6-nbpt-52884-2753552/> (дата обращения: 16.04.2024).
13. Преобразователь скорости потока воздуха для применения в системах ОВК // URL: <https://www.epluse.com/products/air-velocity-instrumentation/air-flow-transmitters-and-probes/ee650/> (дата обращения: 17.04.2024).
14. Датчик влажности и температуры с аналоговыми выходами // URL: <https://www.epluse.com/> (дата обращения: 17.04.2024).
15. Жидкокристаллический индикатор буквенно-цифровой 8 символов 2 строки // URL: <https://www.melt.com.ru/shop/displei/segmentnye-zhk-indikatory.html> (дата обращения: 25.04.2024).
16. Инструкция по охране труда монтажника электрооборудования // URL: <http://vseokraskah.net/instrukcii-po-oxrane-truda/instrukciya-po-oxrane-truda-montazhnika-elektrooborudovaniya.html> (дата обращения: 15.05.2024).

ПРИЛОЖЕНИЕ А

(обязательное)

Задание на выпускную квалификационную работу магистра

«Утверждаю»
Заведующий кафедрой
Пронин И.А.
(фамилия, имя, отчество)
«22» мая 2024 г.

ЗАДАНИЕ

на выпускную квалификационную работу магистра

1. Студент гр. 22ПДм1 факультета информационных технологий и электроники
Направление 11.04.04 Электроника и нанoeлектроника

Магистерская программа Нано- и микроэлектроника

Ганин Илья Игоревич

(фамилия, имя, отчество)

2. Руководитель Аношкин Юрий Владимирович

(фамилия, имя, отчество)

3. Тема выпускной работы Система автоматизированного управления
газодинамическим стендом

Тема утверждена приказом ПГУ № 1059/о от «30» 11 2022 г.

4. Техническое задание на выпускную работу (разработать, исследовать (назначение
объекта разработки, исследования), исходные данные, граничные условия
разрабатываемого или исследуемого объекта, условия эксплуатации и т.д.):

Разработать автоматизированную систему управления газодинамическим стендом для
создания парогазовой смеси с заданной концентрацией контрольного компонента.

Характеристики стенда:

– температура термостата источника парогазовой среды: от 20 до 99 °C;

– сброс излишней парогазовой среды: от 0 до 99 %;

– разбавление воздухом парогазовой среды: от 0 до 30 л/мин;

– влажность разбавляемого воздуха: от 10 до 90 %.

5. Объем и содержание основной части выпускной работы

5.1. Пояснительная записка (перечень вопросов, подлежащих разработке, обоснований,
описаний, расчетов и т.д.)

1. Обзор газодинамических систем и компонентов автоматизации

1.1 Газодинамический стенд

1.2 Шаговый двигатель компонент автоматизации

1.3 Типы температурных датчиков и принцип их работы

1.4 Газодинамические датчики потока

1.5 Типы датчиков влажности и принцип их работы

2. Разработка системы автоматизированного управления газодинамическим стендом

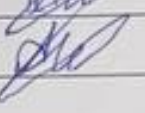
2.1 Микропроцессорная техника в системах управления технологическими процессами

2.2 Язык программирования

2.3 Микроконтроллеры серии 8051

2.4 Система автоматизированного управления газодинамическим стендом	
2.5 Разработка модуля управления температурой	
2.6 Разработка модуля управления сбросом парогазовой смеси	
2.7 Разработка модуля управления разбавлением воздухом	
2.8 Разработка модуля управления влажностью воздуха	
2.9 Разработка главного модуля управления	
2.10 Автоматизированная работа газодинамического стенда	
3. Техника безопасности при эксплуатации электрооборудования	
5.2. Иллюстративная часть (перечень и содержание слайдов презентации)	
1. Тема ВКР и информация об авторе, слайд №1	
2. Цель и задачи выпускной квалификационной работы, слайд №2	
3. Газодинамический стенд, слайд №3	
4. Функциональная схема автоматизированной системы управления, слайд №4	
5. Схема электрическая принципиальная, слайд №5	
6. Моделирование работы модуля управления температурой, слайд №6	
7. Моделирование работы модуля управления сбросом парогазовой смеси, слайд №7	
8. Моделирование работы модуля управления разбавлением воздухом, слайд №8	
9. Моделирование работы модуля управления влажностью воздуха, слайд №9	
10. Моделирование работы главного модуля управления, слайд №10	
11. Выводы по ВКР; слайд №11	
12. Заключительный слайд с контактными данными; слайд №12	
6. Консультанты и содержание дополнительных разделов	
6.1. _____	
Консультант _____	

7. Календарный график по выполнению работы магистра

Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя, консультанта
Анализ компонентов электроники используемых в системах автоматизации	10%	23.05.2024	
Анализ применяемых измерительных приборов используемых в системе автоматизации	10%	25.05.2024	
Анализ исполнительных компонентов применяемых в качестве движущей силы в системах автоматизации	15%	26.05.2024	
Анализ микропроцессорной техники, которые упрощают систему управления технологическим процессом	10%	27.05.2024	
Разработка автоматизированной системы управления газодинамическим стендом	40%	30.05.2024	
Графическая часть.	5%	31.05.2024	
Оформление пояснительной записки.	10%	1.06.2024	

Руководитель работы магистра

подпись

Аношкин Ю.В.

фамилия, инициалы

Дата выдачи задания

«22»

мая

2024 г.

Задание к исполнению принял

подпись

Ганин И.И.

фамилия, инициалы

«22»

мая

2024 г.

Работу магистра к защите допустить

Декан факультета

подпись

Кревчик В.Д.

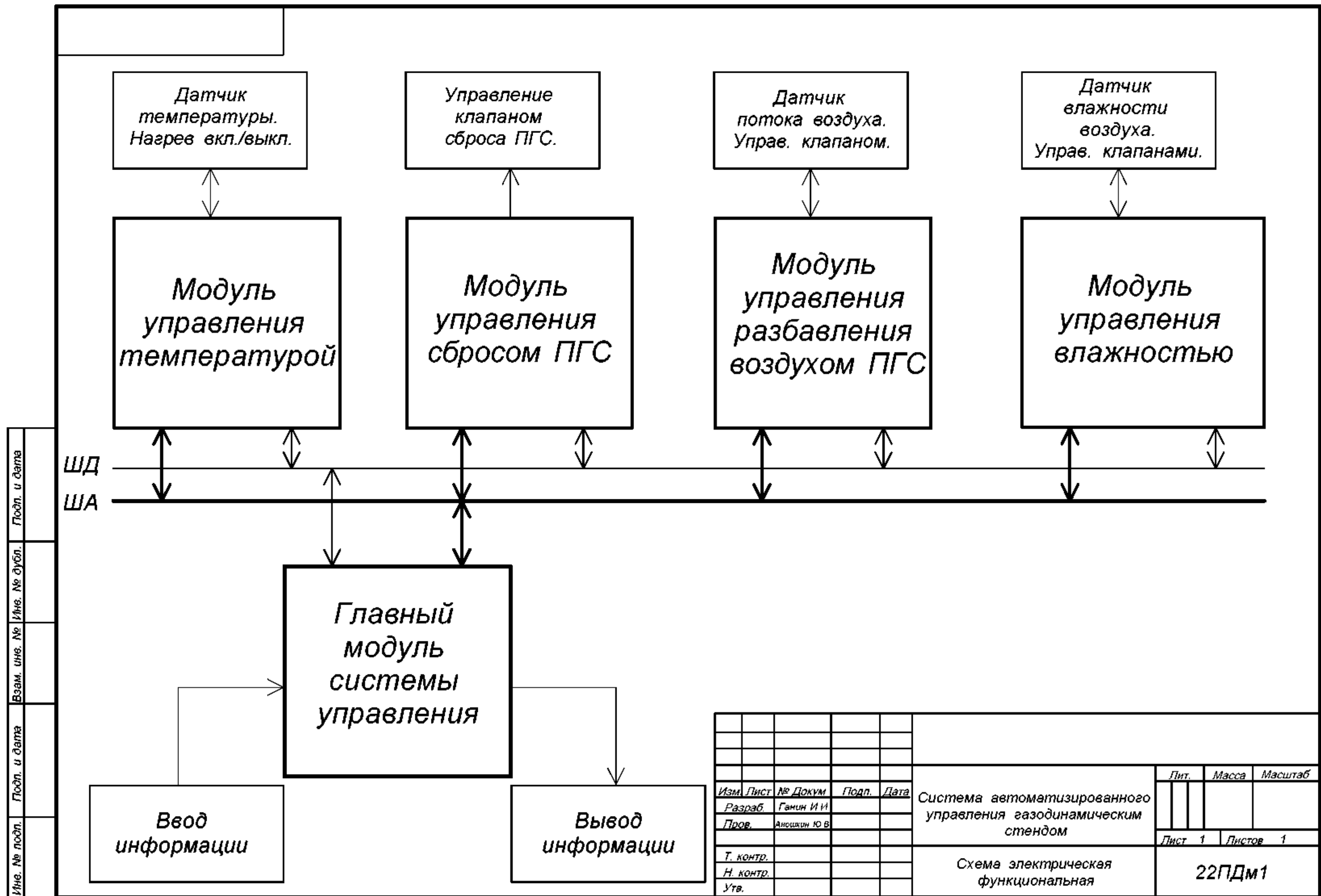
фамилия, инициалы

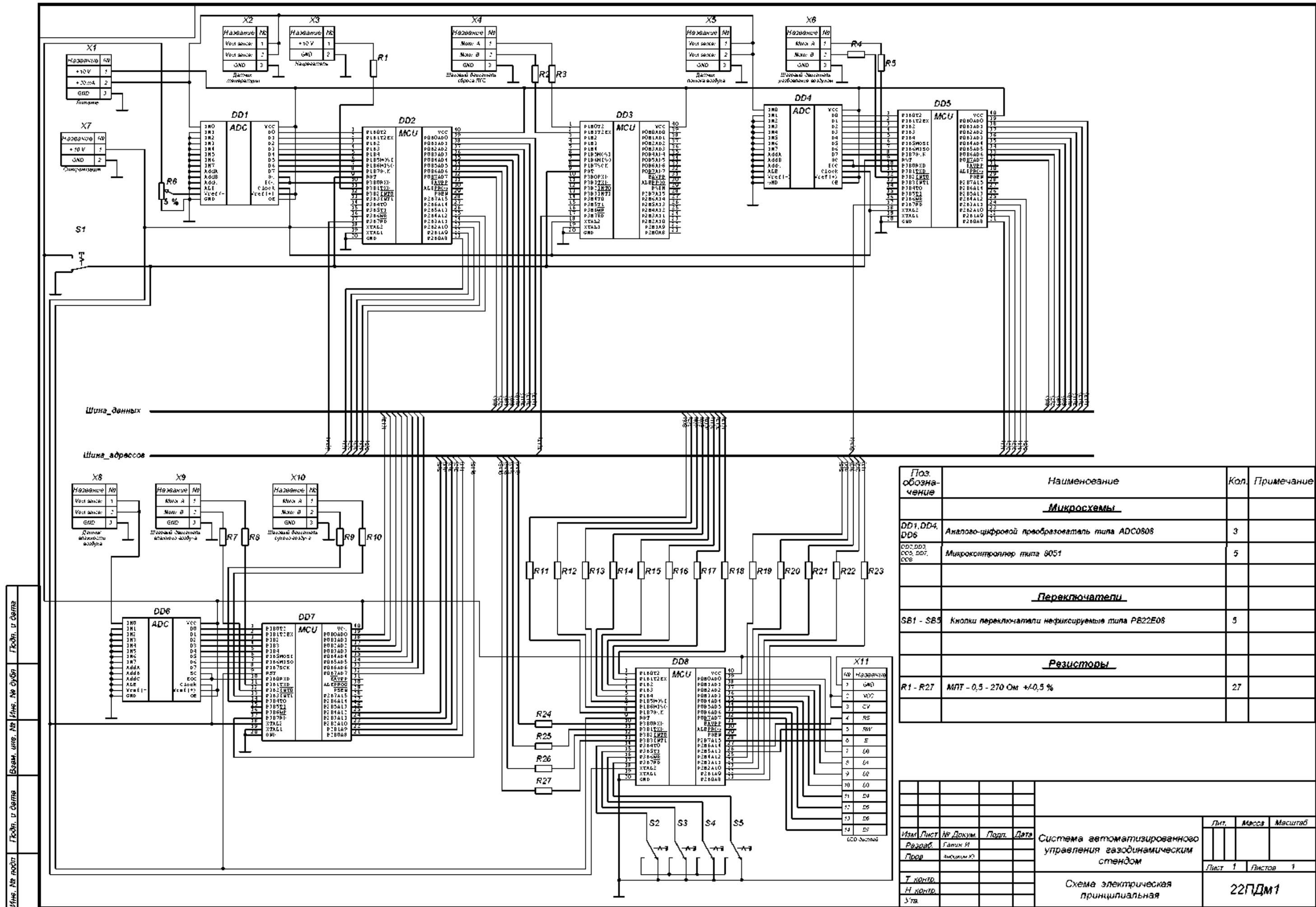
«18» июня 2024 г.

ПРИЛОЖЕНИЕ Б

(обязательное)

Схема электрическая функциональная, схема электрическая принципиальная





ПРИЛОЖЕНИЕ В

(обязательное)

Листинг программ микроконтроллеров 5081

```

Программа микроконтроллера для модуля управления температурой
/*****
*      Module program control temperature in GDS *
*****/

#include<8051.h>                                // On module 8051

//      CONTENT
void Init();                                // Subprogram initialization
void Run();                                // Subprogram run
void End();                                // Subprogram end
void msec(int x);                            // Subprogram pause
unsigned char Measuring(unsigned char port,
                        unsigned char param);    //Function measuring of temperature
unsigned char Set_param(unsigned char param);    //Function installparameter of
temperature
unsigned char Sent_measury(unsigned char param); //Function sent measure temperature

// Base program
void main()
{
    Init();
    Run();
    End();
}

// ===== Sub programs =====

// 1. Subprogram initialization
void Init(){
    unsigned char time_out = 100;
    P0 = 0xFF;                                // Input port P0
    P1 = 0xFF;                                // Input port P1
    P2 = 0xFF;                                // Input port P2
    P3 = 0x00;                                // Output port P3

    msec(time_out);
    P3=1;
    while(P24==0);
    P3=0;
}

// 2. Subprogram run
void Run(){
    unsigned char install_param = 128;        // Temperature in hex (65 'C)
    unsigned char send_param = 0;            // Measure temperature and sent in main
    unsigned char Port_0;                     // Data port P0 (In/Out)
    unsigned char Port_1;                     // Input measuretemperature
    unsigned char Port_2;                     // Input Commands P2: Cmd_T(P20)
    Cmd_S(P21) Cmd_R(P22) Cmd_RH(P23) Cmd_Sent(P24)

```

```

Port_2 = P2;                                     // Port for commands subprogram

while((P20 && P21 && P22 && P23 && P24)==0){      // Exit Run

    if((P20 == 1) && (P24 == 1) && (P21 == 0) && (P22 == 0) && (P23 == 0)) { //
Cmd_T(P20)=1 and Cmd_Sent(P24)=1
        P0=0xFF;
        Port_0 = P0;
        install_param = Set_param(Port_0);
        while((P20 == 1) && (P24 == 1)) P37 = 1;
        P37 = 0;
    }

    // Measuring temperature
    Port_1 = P1;
    send_param = Measuring(Port_1, install_param);

    // Sent temperature in main
    if((P20 == 1) && (P24 == 0) && (P21 == 0) && (P22 == 0) && (P23 == 0)){ //
Cmd_T(P20)=1 and Cmd_Sent(P24)=0
        ACC = Sent_measury(send_param);    // Record in accumulator

        while (P20 == 1){
            P0 = ACC;                        // Output port P0
            P37=1;                          // Cmd_T_sent On
        }

        P37=0;                             // Cmd_T_sent Off
        P0=0xFF;

    }
}

// 3. Subprogram end
void End(){
    P31 = 0;                                // Off temperature
    while((P20 || P21 || P22 || P23 || P24)!=0);
}

// 4. Subprogram pause x*10 msec
void msec(int x){
    while(x-- > 0){
        TH0 = (-10000) >> 8;
        TL0 = (-10000);
        TR0=1;
        do; while(TF0 == 0);

        TF0 = 0;
        TR0 = 0;
    }
}

```

```
// 5. Function measuring temperature
//      where      port - measureparameter temperature, hex
//                  param - install parameter temperature, hex
unsigned char Measuring(unsigned char port, unsigned char param){
unsigned char rezult=0;
```

```
    P30 = 1;
    rezult = port;
    if (rezult<param) P31 = 1;
    else P31 = 0;
    P30 = 0;
```

```
    returnrezult;
}
```

```
// 6. Function installparameter temperature
//      where set_param - install parameter temperature, 'C
unsigned char Set_param(unsigned char set_param){
    // hex_T = a*100*param + b*100
    unsigned char a = 196;           // a*100
    unsigned char b = 44;            // b*100
    unsignedint buff = 0;
    unsigned char rezult = 0;
```

```
    buff = (a * set_param + b)/100;
    rezult = (unsigned char) buff;
    returnrezult;
}
```

```
// 7. Subprogram sent measure temperature
//      where param - measureparameter temperature, hex
unsigned char Sent_mesury(unsigned char param){
    // T_'C = a*100*param - b*100
    unsigned char a = 51;           // a*100
    unsigned char b = 23;           // b*100
    unsignedint buff = 0;
    unsigned char rezult = 0;
```

```
    buff = (a * param - b)/100;
    rezult = (unsigned char) buff;
    P0=0x00;
    returnrezult;
```

```
}
```

```

Программа микроконтроллера для модуля управления сбросом избыточной ПГС
/*****
*      Module program control shedding in GDS *
*****/

#include<8051.h>                                // Onmodule 8051

typedefstruct {
    intpoz_proc;                                // Per cent open Valve, %
    intsig_i;                                    // Last signal motor, (00, 10, 11, 01)
}Valve;                                          // My type <Valve> for motor

//      CONTENT
voidInit();                                    // Subprogram init
void Run(Valve *pz);                           // Subprogram run
void End(Valve *pz);                           // Subprogram end
voidmsec(int x);                               // Subprogram pause

voidInstallValve(unsigned char inst,            // Function install position valve
                 unsigned char pos,
                 Valve *p );
unsigned char Rotation(unsigned char step,       // Function rotation motor
                      unsigned char left_right,
                      int k);

// Base program
void main()
{
    Valve poz;
    int *pz;
    pz = &poz;

    Init();
    Run(pz);
    End(pz);
}

// ===== Sub programs =====

// 1. Subprogram initialization
voidInit(){
    unsigned char time_out = 100;

    P0 = 0x00;                                // Output port P0
    P1 = 0xFF;                                // Input port P1
    P2 = 0xFF;                                // Input port P2
    P3 = 0x00;                                // Output port P3

    msec(time_out);
    P37=1;

```

```

while(P24==0);
    P37=0;
}

// 2. Subprogram run
void Run(Valve *pz){
    Valve vlv; //Type for motor <Valve>
    Valve *v; // Reference to type <Valve>
    unsigned char install_param = 1; // Shedding in % (50%)
    unsigned char position_valve = 0; // Position valve real
    unsigned char position_valve_i = 0; // Last signal motor
    unsigned char send_param = 0; // Measure shedding and sent in main
    unsigned char Port_0; // Data port P0 (In/Out)
    unsigned char Port_2; // Input Commands P2: Cmd_T(P20)
    Cmd_S(P21) Cmd_R(P22) Cmd_RH(P23) Cmd_Sent(P24)

    v = &vlv;
    Port_2 = P2; // Port for commands subprogram
    vlv.poz_proc=(int) position_valve;
    vlv.sig_i = (int) position_valve_i;

    while((P20 && P21 && P22 && P23 && P24)==0){ // Exit Run

        if((P21 == 1) && (P24 == 1)&& (P20 == 0)&& (P22 == 0)&& (P23 == 0)) { //
            Cmd_S(P21)=1 and Cmd_Sent(P24)=1
            P0 = 0xFF;
            Port_0 = P0;
            install_param = Port_0;

            if(install_param< 100){
                InstallValve(install_param, position_valve, v); // Install valve for motor
                position_valve = vlv.poz_proc;
                position_valve_i = vlv.sig_i;
            }

            while((P21 == 1) && (P24 == 1)) P37 = 1;
            P37 = 0;
        }

        // Sent shedding in main
        if((P21 == 1) && (P24 == 0)&& (P20 == 0)&& (P22 == 0)&& (P23 == 0)){ //
            Cmd_S(P21)=1 and Cmd_Sent(P24)=0
            P0 = 0x00;
            ACC = position_valve; // Record in accumulator

            while (P21 == 1){
                P0 = ACC; // Output port P0
                P37 = 1; // Cmd_S_sent On
            }

            P37 = 0; // Cmd_S_sent Off
            P0 = 0xFF;

```

```

    }
}

pz->poz_proc = v->poz_proc;
pz->sig_i = v->sig_i;
}

// 3. Subprogram end
void End(Valve *pz){
int position;

position = pz->poz_proc;
InstallValve(1, position, pz);
P37 = 1;

while((P20 || P21 || P22 || P23 || P24)!=0);
}

// 4. Subprogram pausa x*10 msec
voidmsec(int x){
while(x-- > 0){
    TH0 = (-10000) >> 8;
    TL0 = (-10000);
    TR0 = 1;
do; while(TF0 == 0);

    TF0 = 0;
    TR0 = 0;
}
}

//5. Function install position valve
// where      inst - install position;
//            pos - real position.

voidInstallValve(unsigned char inst, unsigned char pos, Valve *p){
int i = 0;
unsigned char steps = 0;
unsigned char left_right = 0;
unsigned char d_prc = 0;           // |ins - pos|
unsigned char k_prc_steps = 3;    // Coefficient transform proc. to steps.

if (pos == inst) return;

    i = p->sig_i;

if (pos<inst) {
left_right = 0;
d_prc = inst - pos;
}

if (pos>inst) {

```

```

    left_right = 1;
    d_prc = pos - inst;
    }

    i = (int) d_prc* k_prc_steps;
    steps = (unsigned char) i;
    P1 = 0x00;

    k_prc_steps = Rotation(steps, left_right, i);
    i = (int) k_prc_steps;
    p->poz_proc = inst;
    p->sig_i = i;
    }

// 6. Function rotation motor
// where      step - all steps;
//            left_right - rotation left=0 - open or right=1 - close.
unsigned char Rotation(unsigned char step, unsigned char left_right, int k){
    unsigned char result = 0;
    unsigned char j = 0;
    int i = 0;
    intsteps_cl[4] = {0b00000000,
                      0b00000001,
                      0b00000011,
                      0b00000010}; // Close valve
    intsteps_op[4] = {0b00000010,
                      0b00000011,
                      0b00000001,
                      0b00000000}; // Open valve

    i = k;
    do{
        if (left_right == 0){
            if (i > 3) i = 0;
            P1 = steps_cl[i];
            result = (unsigned char) i;
            i++;
        }
        else{
            if (i > 3) i = 0;
            P1 = steps_op[i];
            result = (unsigned char) i;
            i++;
        }
        j++;
    }while(j <= (step));

    return result;
}

```

Программа микроконтроллера модуля управления разбавления воздухом ПГС
 /*****

* Moduleprogram control doping in GDS *
 *****/

#include<8051.h> // Onmodule 8051

```
typedefstruct {
intpoz_proc;                               // Per cent open Valve, %
intsig_i;                                 // Last signl motor, (00, 10, 11, 01)
}Valve;                                 // My type <Valve> for motor
```

CONTENT

```
voidInit();                               // 1. Subprogram initialization
void Run(Valve *pm);                    // 2. Subprogram run
void End(Valve *pm);                    // 3. Subprogram end
voidmsec(int x);                        // 4. Subprogram pause
unsigned char Set_param(unsigned char set_param, // 5. Function installparameter
doping
Valve *pm);
unsigned char Sent_measury(unsigned char param); // 6. Subprogram sent measure
doping
unsigned char Rotation(unsigned char step,                // 7. Function rotation motor
unsigned char left_right, int k);
```

// Base program

```
void main(){
Valve motor;
Valve *m;
m = &motor;
m->poz_proc=0;
m->sig_i=0;
```

```
Init();
Run(m);
```

```

End(m);
}

// ===== Sub programs =====

// 1. Subprogram initialization
void Init(){
unsigned char time_out = 100;

    P0=0xFF;           // Input port P0
    P1=0xFF;           // Input port P1
    P2=0xFF;           // Input port P2
    P3=0x00;           // Output port P3

    msec(time_out);
    P3=1;
    while(P2==0);
    P3=0;
}

// 2. Subprogram run
void Run(Valve *pm){
unsigned char install_param = 128;           // Doping in hex (65 'C)
unsigned char send_param = 0;               // Measure doping and sent in main
unsigned char Port_0;                       // Data port P0 (In/Out)
unsigned char Port_1;                       // Input measure doping
unsigned char Port_2;                       // Input Commands P2: Cmd_T(P20)
Cmd_S(P21) Cmd_R(P22) Cmd_RH(P23) Cmd_Sent(P24)

    Port_2 = P2;                           // Port for comands subprogram

    while((P20 && P21 && P22 && P23 && P24)==0){           // Exit Run

```

```

        if((P22 == 1) && (P24 == 1) && (P20 == 0) && (P21 == 0) && (P23 == 0)) { //
Cmd_R(P22)=1 and Cmd_Sent(P24)=1
        P0=0xFF;
        Port_0 = P0;
install_param = Set_param(Port_0, pm);
while((P22 == 1) && (P24 == 1)) P37 = 1;
        P37 = 0;
    }

    // Measuring doping
    P30 = 1;
    Port_1 = P1;
    P30 = 0;

send_param = Port_1;

    // Sent doping in main
    if((P22 == 1) && (P24 == 0) && (P20 == 0) && (P21 == 0) && (P23 == 0)){ //
Cmd_R(P22)=1 and Cmd_Sent(P24)=0
        ACC = Sent_measury(send_param); //      Record      in
accumulator

        while (P22 == 1){
            P0 = ACC; //
Output port P0
            P37=1; //
Cmd_R_sent On
        }

        P37=0; //
Cmd_R_sent Off
        P0=0xFF;
    }
}
}
}

```

```

// 3. Subprogram end
void End(Valve *pm){
int position = pm ->poz_proc;
int signal = pm ->sig_i;
int step = 18;                // Step (1,8 * 10)
int steps = 0;

steps = position / step * 10;
signal = Rotation(steps, 0, signal);
pm ->poz_proc = 0;
pm ->sig_i = signal;
P37 = 1;

while((P20 || P21 || P22 || P23 || P24)!=0);
}

// 4. Subprogram pausa x*10 msec
voidmsec(int x){
while(x-- > 0){
    TH0 = (-10000) >> 8;
    TL0 = (-10000);
    TR0=1;
do; while(TF0 == 0);

    TF0 = 0;
    TR0 = 0;
}
}

// 5. Function istallparametr doping
//      where set_param - install parametr doping, L/min
//      *pm- reference on step motor Per cent open Valve, %
//      and Last signl motor, (00, 10, 11, 01)

```

```

unsigned char Set_param(unsigned char set_param, Valve *pm){
unsigned char rezult = 0;
unsigned char Port_0 = 0;           // Port in signal measure
intparam_min = 0;
intparam_max = 0;
int steps = 0;                     // All steps
intd_f = 0;                        // Delta doping
int buff = 0;                      // Buffer variable
intd_param = 10;                   // Inaccuracy, %
int step = 18;                     // Step (1,8 * 10)
unsigned char k_f_d = 18;           // Coefficient transform doping L/min(30) to
                                     // deg_max(540)
unsigned char k_hex_L_min = 119; // Coefficient transform hex to L/min
                                     // (k_hex_L_min * 1000)

P30 = 1;                           // Start measure
Port_0 = P0;                       // Input signal from sensor

P30 = 0;                           // Stop measure

buff = (int) set_param - set_param * d_param / 100; // Min set_param
if (buff<=0) param_min = 0;
elseparam_min = buff * 1000 / k_hex_L_min;

buff = (int) set_param + set_param * d_param / 100; // Max set_param
if (buff >= 30) param_max = 30 * 1000 / k_hex_L_min;
elseparam_max = buff * 1000 / k_hex_L_min;

while((Port_0 <= param_min) || (Port_0 >= param_max)){
if ( Port_0 <= param_min){           // Open valve
doping
d_f = param_min - Port_0;
d_param = d_f * k_f_d;
buff = pm->poz_proc - d_param;
pm->poz_proc = buff;

```

```

steps = (int) d_param / step * 10;
buff = pm->sig_i;
buff = Rotation(steps, 0, buff);
pm->sig_i = buff;
if (pm->poz_proc <= 0) return 0;
    }
if (Port_0 >= param_max) {           // Close valve doping
d_f = Port_0 - param_max;
d_param = d_f * k_f_d;
buff = pm->poz_proc + d_param;
pm->poz_proc = buff;
steps = (int) d_param / step * 10;
buff = pm->sig_i;
buff = Rotation(steps, 1, buff);
pm->sig_i = buff;
if (pm->poz_proc >= 540) return 255;
}

P30 = 1;                             // Start measure
Port_0 = P0;                         // Input signal from sensor

P30 = 0;                             // Stop measure
}
result = Port_0;
return result;
}

// 6. Subprogram sent measury doping
//      where param - measuryparametr doping, hex
unsigned char Sent_measury(unsigned char param){
unsigned char k_hex_L_min = 119;      // Coefficient transform hex to L/min
                                       //(k_hex_L_min * 1000)

int buff = 0;
unsigned char result = 0;

```

```

buff = (int) k_hex_L_min * param / 1000;
result = (unsigned char) buff;
    P0=0x00;
returnresult;
}

// 7. Function rotation motor
// where      step - all steps;
//              left_right - rotation left=0 - open or right=1 - close.
unsigned char Rotation(unsigned char step, unsigned char left_right, int k){
unsigned char result = 0;
unsigned char j = 0;
int i = 0;
intsteps_cl[4] = {0b00000000,
                  0b00000010,
                  0b00000110,
                  0b00000100}; // Close valve
intsteps_op[4] = {0b00000010,
                  0b00000110,
                  0b00000010,
                  0b00000000}; // Open valve

i = k;
do{
if (left_right == 0){
if (i > 3) i = 0;
    P3 = steps_cl[i];
result =(unsigned char) i;
i++;
    }
else{
if (i > 3) i=0;
    P3 = steps_op[i];
result = (unsigned char) i;
i++;

```

```

    }
    j++;
    }while(j <= (step));

    returnresult;
}

```

Программа микроконтроллера модуля управления влажностью воздуха
 /*****

* Moduleprogram control humidity in GDS *
 *****/

#include<8051.h> // On module 8051

```

typedefstruct {               // humid
int M1_poz_proc;               // Per cent open Valve M1, %
int M1_sig_i;                // Last signal motor M1, (00, 10, 11, 01)
// arid
int M2_poz_proc;               // Per cent open Valve M2, %
int M2_sig_i;                // Last signal motor M2, (00, 10, 11, 01)
}Valves;                      // Secial type <Valves> for motors

```

// CONTENT

```

voidInit(Valves *pm);        // 1. Subprogram initialisation
void Run(Valves *pm);        // 2. Subprogram run
void End(Valves *pm);        // 3. Subprogram end
voidmsec(int x);              // 4. Subprogram pausa
voidSet_param(unsigned char set_param, Valves *pm);    //        5.        Subprogram
istallparametr humidity
unsigned char Sent_measury(unsigned char param); // 6. Subprogram sent measury
humidity
unsigned char Rotation(unsigned char steps,                // 7. Function rotation motor
                          unsigned char left_right, int k,
                          unsigned char mot);

```

```

// Base program
void main(){
    Valves motors;
    Valves *m;
    m = &motors;
    m->M1_poz_proc=0;          // humid
    m->M1_sig_i=0;              // humid
    m->M2_poz_proc=0;          // arid
    m->M2_sig_i=0;              // arid

    Init(m);
    Run(m);
    End(m);
}

// ===== Sub programs =====

// 1. Subprogram initialization
void Init(Valves *pm){
    unsigned char time_out = 10;
    unsigned char steps = 150;
    P0=0xFF;                    // Input port P0
    P1=0xFF;                    // Input port P1
    P2=0xFF;                    // Input port P2
    P3=0x00;                    // Output port P3

    pm -> M1_sig_i = Rotation(steps, 1, 0, 0);
    pm -> M1_poz_proc = 50;

    pm -> M2_sig_i = Rotation(steps, 1, 0, 1);
    pm -> M2_poz_proc = 50;

    msec(time_out);

```

```

    P37=1;
while(P24==0);
    P37=0;
}

// 2. Subprogram run
void Run(Valves *pm){
    unsigned char install_param = 128;           // Humidity in hex (45 %)
    unsigned char send_param = 0;                // Measure humidity and sent in main
    unsigned char Port_0;                        // Data port P0 (In/Out)
    unsigned char Port_1;                        // Input measure humidity
    unsigned char Port_2;                        // Input Commands P2: Cmd_T(P20)
    Cmd_S(P21) Cmd_R(P22) Cmd_RH(P23) Cmd_Sent(P24)

    Port_2 = P2;                                // Port for comands subprogram

    while((P20 && P21 && P22 && P23 && P24)==0){    // Exit Run

        if((P23 == 1) && (P24 == 1) && (P20 == 0) && (P21 == 0) && (P22 == 0)) { //
            Cmd_R(P22)=1 and Cmd_Sent(P24)=1
            P0=0xFF;
            Port_0 = P0;
            Set_param(Port_0, pm);
            while((P23 == 1) && (P24 == 1)) P37 = 1;
            P37 = 0;
        }

        // Measuring humidity
        P30 = 1;
        Port_1 = P1;
        P30 = 0;

        send_param = Port_1;

        // Sent humidity in main

```

```

if((P23 == 1) && (P24 == 0) && (P20 == 0) && (P21 == 0) && (P22 == 0)){
    // Cmd_RH(P23) and Cmd_Sent(P24)=0
    ACC = Sent_measury(send_param); // Record in accumulator

while (P23 == 1){
    P0 = ACC; // Output port P0
    P37=1; // Cmd_RH_sent On
    }

    P37=0; // Cmd_RH_sent Off
    P0=0xFF;
    }
    }
}

// 3. Subprogram end
void End(Valves *pm){
    int step = 18; // Step (1,8 * 10)
    int steps = 0;

    steps = pm -> M1_poz_proc / step * 10;
    pm -> M1_sig_i = Rotation(steps, 0, pm -> M1_sig_i, 0);
    pm -> M1_poz_proc = 0;

    steps = pm -> M2_poz_proc / step * 10;
    pm -> M2_sig_i = Rotation(steps, 0, pm -> M2_sig_i, 1);
    pm -> M2_poz_proc = 0;

    P37 = 1;

while((P20 || P21 || P22 || P23 || P24)!=0);
}

// 4. Subprogram pausa x*10 msec

```

```

void msec(int x){
while(x-- > 0){
    TH0 = (-10000) >> 8;
    TL0 = (-10000);
    TR0=1;
do; while(TF0 == 0);

    TF0 = 0;
    TR0 = 0;
}
}

// 5. Subprogram install parametr humidity
//      where set_param - install parametr humidity, %
//      *pm- reference on step motor Per cent open Valve, %
//      and Last signal motor, (00, 10, 11, 01)
void Set_param(unsigned char set_param, Valves *pm){
unsigned char result = 0;
unsigned char Port_1 = 0;          // Port in signal measure
unsigned char set = 0;
unsigned char set_min = 0;
unsigned char set_max = 0;
int steps = 0;
unsigned char d_set = 30;
unsigned char k_rh_hex = 28;      // Coefficient transform rh(%) to hex, (2.8*10)
unsigned char k_hex_steps = 117;  // Coefficient transform hex to steps, (1.17*100)
int step = 18;                   // Step (1,8 * 10)

set = set_param * k_rh_hex /10;
set_min = set - set * d_set /100;
set_max = set + set * d_set /100;

P30 = 1;                        // Start measure
Port_1 = P1;                     // Input signal from sensor

```

```

P30 = 0;                                // Stop measure

while((Port_1 <= set_min) || (Port_1 >= set_max)){
if(Port_1 <= set_min){
d_set = (unsigned char) (set_min - Port_1) / 2;
steps = (int)d_set * k_hex_steps / 100;
pm -> M1_sig_i = Rotation(steps, 0, pm -> M1_sig_i, 0);
pm -> M2_sig_i = Rotation(steps, 1, pm -> M2_sig_i, 1);
    }

if(Port_1 >= set_max){
d_set = (unsigned char) ( Port_1 - set_max) / 2;
steps = (int)d_set * k_hex_steps / 100;
pm -> M1_sig_i = Rotation(steps, 1, pm -> M1_sig_i, 0);
pm -> M2_sig_i = Rotation(steps, 0, pm -> M2_sig_i, 1);
    }

    P30 = 1;                                // Start measure
    Port_1 = P1;                            // Input signal from sensor
    P30 = 0;                                // Stop measure
}
}

// 6. Subprogram sent measury humidity
//      where param - measuryparametr humidity, hex
unsigned char Sent_measury(unsigned char param){
unsigned char k_rh_hex = 28;                // Coefficient transform rh(%) to hex, (2.8*10)
int buff = 0;
unsigned char rezult = 0;

buff = (int) param / k_rh_hex *10;
rezult = (unsigned char) buff;
    P0=0x00;
returnrezult;
}

```

```

// 7. Function rotation motor
// where      steps - all steps;
//            left_right - rotation, left=0 - open or right=1 - close;
//            mot - select step motor, mot=0 - M1 or mot=1 - M2
unsigned char Rotation(unsigned char steps, unsigned char left_right, int k, unsigned char
mot){
    unsigned char result = 0;
    unsigned char j = 0;
    int i = 0;
    intstep_cl[4][2] = {0, 1, 1, 0, // Close valve
                        0, 0, 1, 1};

    intstep_op[4][2] = {0, 1, 1, 0,      // Open valve
                        1, 1, 0, 0};

    i = k;
    do{
        if (left_right == 0){
            if (i > 3) i = 0;
            if (mot == 0){
                P31 = step_op[i][0];
                P32 = step_op[i][1];
            }
            else {
                P33 = step_op[i][0];
                P34 = step_op[i][1];
            }
            result =(unsigned char) i;
            i++;
        }
        else{
            if (i > 3) i=0;
            if (mot == 0){

```

```

        P31 = step_cl[i][0];
        P32 = step_cl[i][1];
    }
else {
    P33 = step_cl[i][0];
    P34 = step_cl[i][1];
}
result = (unsigned char) i;
i++;
}
j++;
}while(j <= (steps));

returnresult;
}

```

Программа микроконтроллера главного модуля управления ГДС
 /*****

* Moduleprogram control parameters GDS *
 *****/

#include<8051.h> // Onmodule 8051

```

typedefstruct {
unsigned char temp;
unsigned char rh;
unsigned char shed;
unsigned char dop;
}Param;

```

 // CONTENT

```

voidInit();                 // 1. Subprogram initialization
void Run();                // 2. Subprogram run
void End();                // 3. Subprogram end
voidmsec(int x);           // 4. Subprogram pause

```

```

void Print(unsigned char msg[]);    // 5. Subprogram print message on the LCD
int Input(Param *m);               // 6. Subprogram input parameters for install.
intAll_Measuring(Param *m);        // 7. All measuresparameters
charDigChar(int n);                // 8. Translate int to char
voidSend_param(Param *m);          // 10. Send parameter GDS

```

```

// Base program

```

```

void main(){
Init();
Run();
End();
}

```

```

// ===== Sub programs =====

```

```

// 1. Subprogram initialization

```

```

voidInit(){
unsigned char welcom[16] = "GDS ready ";

```

```

P0 = 0x00;    // Output port P0
P1 = 0xFF;    // Input port P1
P2 = 0x00;    // Output port P2
P3 = 0xFF;    // Input port P3

```

```

Print(welcom);

```

```

P24 = 1;

```

```

while ((P34 && P35 && P36 && P37) == 0){    //(P34 && P35 && P36 && P37) ==

```

0

```

    P24 = 1;
}

```

```

P24 = 0;

```

```

}

```

```

// 2. Subprogram run
void Run(){
    intflag_run = 1;
    Parammesury;
    Param *pr;
    pr = &mesury;

    while(flag_run == 1){
        flag_run = All_Measuring(pr);
        if (P30 == 1) {
            flag_run = Input(pr);
            Send_param(pr);
        }
    }
}

// 3. Subprogram end
void End(){
    unsigned char off[16] = "OFF   Stend  ";

    Print (off);
    while((P34 && P35 && P36 && P37)== 1);
}

// 4. Subprogram pausa x*10 msec
voidmsec(int x){
    while(x-- > 0){
        TH0 = (-10000) >> 8;
        TL0 = (-10000);
        TR0=1;
        do; while(TF0 == 0);

        TF0 = 0;
        TR0 = 0;
    }
}

```

```
}
```

```
// 5. Subprogram print message on the LCD, msg - array text?n-all char
```

```
void Print(unsigned char msg[]){
```

```
    unsigned char n = 16;
```

```
    unsigned char i = 0;
```

```
    P26 = 1; // Output LCD (RS)
```

```
    P27 = 1;
```

```
    for (i = 0; i < n; i++){
```

```
        P27 = 0;
```

```
        P0 = msg[i];          // Output message 'msg' on the LCD
```

```
        P27 = 1;
```

```
    }
```

```
    P26 = 0;    // Out LCD (RS)
```

```
}
```

```
// 6. Subprogram input parameters for install.
```

```
int Input(Param *m){
```

```
    intd_par[4] = {10, 5, 15, 1};
```

```
    intd_p_max[4] = {80, 90, 99, 30};
```

```
    int par[4] = {0, 0, 0, 0};
```

```
    unsigned char lcd_set[16] = "      Set Y: ";
```

```
    unsigned char set = 0;
```

```
    unsigned char entr = 0;
```

```
    unsigned char plus = 0;
```

```
    unsigned char minus = 0;
```

```
    unsigned char buff = 0;
```

```
    unsigned char buff_old = 0;
```

```
    int buff_2 = 0;
```

```
    int menu = 0;
```

```
    int exit = 0;
```

```
int exit_2 = 0;

par[0] = (int) m->temp;
par[1] = (int) m->rh;
par[2] = (int) m->shed;
par[3] = (int) m->dop;
```

```
while(exit == 0){
    set = 0;
    entr = 0;
    plus = 0;
    minus = 0;
```

```
    switch(menu){
    case 0:
        lcd_set[12] = 'T';
        break;
```

```
    case 1:
        lcd_set[12] = 'H';
        break;
```

```
    case 2:
        lcd_set[12] = 'S';
        break;
```

```
    case 3:
        lcd_set[12] = 'D';
        break;
```

```
    default:
        break;
    }
```

```

Print(lcd_set);

while((set || entr || plus || minus) == 0 ){
set = P30;
entr = P31;
plus = P32;
minus = P33;
    }

    // Exit install parameters
    if (set == 1) {
m->temp = (unsigned char) par[0];
m->rh = (unsigned char) par[1];
m->shed = (unsigned char) par[2];
m->dop = (unsigned char) par[3];
return 1;
exit = 1;
break;
    }

    // Select parametr ++
    if (plus == 1) {
menu++;
if (menu > 3) menu = 0;
    }

    // Select parametr --
    if (minus == 1) {
menu--;
if (menu < 0) menu = 3;
    }

    // Enter parameter
    if (entr == 1){
lcd_set[0] = 'e';

```

```

buff_old = (unsigned char) par[menu];
buff = buff_old;
    buff_2 = (int) buff / 10;
lcd_set[2] = DigChar(buff_2);

    buff_2 = (int) buff - buff_2 * 10;
lcd_set[3] = DigChar(buff_2);
Print(lcd_set);

    exit_2 = 0;
set = 0;
entr = 0;
plus = 0;
minus = 0;
buff = par[menu];

while(exit_2 == 0){

while((set || entr || plus || minus) == 0){
set = P30;
entr = P31;
plus = P32;
minus = P33;
    }

if (set == 1){
lcd_set[0] = ' ';
lcd_set[2] = ' ';
lcd_set[3] = ' ';
par[menu] = (int) buff_old;
    exit_2 = 1;
break;
    }

if (entr == 1){

```

```

    lcd_set[0] = ' ';
    lcd_set[2] = ' ';
    lcd_set[3] = ' ';
    par[menu] = (int) buff;
        exit_2 = 1;
    break;
    }

    if (plus == 1) {
    buff = (unsigned char) buff + d_par[menu];
    if (buff > d_p_max[menu]) buff = (unsigned char) d_p_max[menu];
        }

    if (minus == 1) {
    buff = (unsigned char) buff - d_par[menu];
    if (buff < 0) buff = 0;
        }

    par[menu] = (int) buff;

        buff_2 = (int) buff / 10;
    lcd_set[2] = DigChar(buff_2);

        buff_2 = (int) buff - buff_2 * 10;
    lcd_set[3] = DigChar(buff_2);

    Print(lcd_set);

    set = 0;
    entr = 0;
    plus = 0;
    minus = 0;
        }
    }
}

```

```

return 1;
}

// 7. All measuresparameters
intAll_Measuring(Param *m){
unsigned char lcd_t_rh[16] =      "T,'C XX RH,% YY ";
unsigned char lcd_sh_dp[16] = "Sh,% XX D,L/mYY ";
    unsigned char value[10] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
unsigned char time_out = 100;          // Pause between output on LCD
measure, msec
    unsigned char buff = 0;

    P1 = 0xFF;          // Input port P1

    if ((P32 && P33 ) == 1) return 0;    //Exit and off

    if (P30 == 1) return 1;          //Exit and set parameters

    // Call temperature
    P20 = 1;
    while(P34 == 0);
    m->temp = P1;
    if(m->temp > 99) m->temp = 99;
    P20 = 0;

    // Call shedding
    P21 = 1;
    while(P35 == 0);
    m->shed = P1;
    if(m->shed > 99) m->shed = 99;
    P21 = 0;

    // Call doping
    P22 = 1;

```

```

while(P36 == 0);
m->dop = P1;
if(m->dop > 30) m->dop = 30;
    P22 = 0;

// Call RH
    P23 = 1;
while(P37 == 0);
m->rh = P1;
if(m->rh > 90) m->rh = 90;
    P23 = 0;

// Format dates
// Temperature
buff = (unsigned char) m->temp / 10;
lcd_t_rh[5] = value[buff];
buff = m->temp - buff * 10;
lcd_t_rh[6] = value[buff];

// RH
buff = (unsigned char) m->rh / 10;
lcd_t_rh[13] = value[buff];
buff = m->rh - buff * 10;
lcd_t_rh[14] = value[buff];

// Shedding
buff = (unsigned char) m->shed / 10;
lcd_sh_dp[5] = value[buff];
buff = m->shed - buff * 10;
lcd_sh_dp[6] = value[buff];

// Doping
buff = (unsigned char) m->dop / 10;
lcd_sh_dp[13] = value[buff];
buff = m->dop - buff * 10;

```

```
lcd_sh_dp[14] = value[buff];
```

```
// Out LCD
```

```
Print(lcd_t_rh);
```

```
msec(time_out);
```

```
Print(lcd_sh_dp);
```

```
msec(time_out);
```

```
return 1;
```

```
}
```

```
// 8. Translate int to char
```

```
charDigChar(int n){
```

```
    unsigned char number[10] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
```

```
    if (n > 9) n = 9;
```

```
    if (n < 0) n = 0;
```

```
    return number[n];
```

```
}
```

```
//10. Send parameter GDS
```

```
voidSend_param(Param *m){
```

```
    unsigned char buff = 0;
```

```
    P1 = 0x00;
```

```
    do{
```

```
        P24 = 1;
```

```
        P20 = 1;
```

```
        P1 = m->temp;
```

```
    }while(P34 == 0);
```

```
P24 = 0;
```

```
P20 = 0;
```

```

do{
    P24 = 1;
    P21 = 1;
    P1 = m->shed;
}while(P35 == 0);

P24 = 0;
P21 = 0;

do{
    P24 = 1;
    P22 = 1;
    P1 = m->dop;
}while(P36 == 0);

P24 = 0;
P22 = 0;

do{
    P24 = 1;
    P23 = 1;
    P1 = m->rh;
}while(P37 == 0);

P24 = 0;
P23 = 0;

P1 = 0xFF;
}

```

ПРИЛОЖЕНИЕ Г

(обязательное)

Заявление и протокол проверки магистерской диссертации на
оригинальность

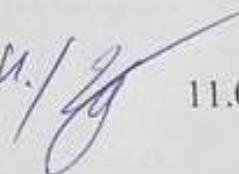
Ректору Пензенского
государственного университета
А.Д. Гулякову
студента 2 курса ФИТЭ
очной формы обучения
Ганина Ильи Игоревича

ЗАЯВЛЕНИЕ

Я, Ганин Илья Игоревич, студент 2 курса очной формы обучения направления подготовки 11.04.04 Электроника и нанoeлектроника, магистерская программа «Нано- и микроэлектроника», прошу провести проверку с использованием системы «Антиплагиат» выпускной квалификационной работы на тему «Система автоматизированного управления газодинамическим стендом», выполненной мной самостоятельно, на содержание элементов плагиата.

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

Я ознакомлен с действующим Стандартом университета «Выпускная квалификационная работа обучающихся по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры», согласно которому обнаружение плагиата является основанием для недопуска ВКР к защите и отчисления из университета.

Ганин И.И. /  11.06.2024

ВКР представлена на проверку 11.06.2024

Руководитель ВКР  Ю.В. Аношкин

проверки на оригинальность в системе «Антиплагиат» магистерской диссертации на тему «Система автоматизированного управления газодинамическим стендом» студента

2 курса И.И. Ганина направления подготовки/специальности 11.04.04,

Электроника и наноэлектроника

профиля/магистерской программы	Нано- и микроэлектроника
--------------------------------	--------------------------

Руководитель ВКР Аношкин Юрий Владимирович



СПРАВКА

© 2006 Blackwell Publishing Ltd, *Journal of Internal Medicine* 260: 459–468

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 103–110

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ ANTIPLAGIAT.RU

Автор работы:	Галин Руслан Игоревич
Самостоятельность работы:	Галин Руслан Игоревич
Название работы:	Система автоматизированного управления гидродинамическим стеном
Тип работы:	Магистерская диссертация
Подразделение:	Кафедра "Механика и микроэлектроника"

РЕЗУЛЬТАТЫ

■ OTHER COMPASSIONATE RESPONSES: While the 1990s have been characterized by a growing emphasis on the importance of the family, the 1990s have also been characterized by a growing emphasis on the importance of the community. This has led to a growing emphasis on the importance of the community in the lives of individuals, and this has led to a growing emphasis on the importance of the community in the lives of individuals.

СОБЛАДЕНИЕ	41,3%	СОБЛАДЕНИЕ	39,93%
ВНЕШНЯЯ БЕЗОПАСНОСТЬ	38,7%	ВНЕШНЯЯ БЕЗОПАСНОСТЬ	39,37%
ЦИТИРОВАННОЕ	0%	ЦИТИРОВАННОЕ	0%
САМОЦИТИРОВАННОЕ	2%	САМОЦИТИРОВАННОЕ	2%

2019年12月15日 星期一

DATA IN PUBLISHED LITERATURE, 1910-2014 1713

Структура
документа
Модуль поиска

Продвижение изделий: титульный лист с 7, оформленная карта с 2-38

[illegible]

Работу проверил: Г.И.Савицкий, к.и.н. А.А.Савицкий, к.и.н.

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 395–402

Дата подписи _____

^a The values are calculated from the following equation: $\text{C}_{\text{max}} = \frac{\text{AUC} \times F}{t_{1/2}}$.



© 2000 Blackwell Science Ltd
Journal of Internal Medicine 247: 399–406
DOI: 10.1046/j.1365-2796.2000.01811.x

Figure 1. The schematic diagram of the proposed system. The system is composed of a user, a server, and a database. The user sends a request to the server, which then queries the database. The server returns the result to the user.

13.01.2004

Ю.В. Аношкин

СПРАВКА

о результатах проверки текстового документа
на наличие заимствований

Пензенский государственный университет

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Ганин Илья Игоревич
Самоцитирование
рассчитано для: Ганин Илья Игоревич
Название работы: Система автоматизированного управления газодинамическим стендом
Тип работы: Магистерская диссертация
Подразделение: Кафедра "Нано- и микроэлектроника"

РЕЗУЛЬТАТЫ

■ ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ: НИЖЕ ПРЕДСТАВЛЕНЫ РЕЗУЛЬТАТЫ ПРОВЕРКИ ДО КОРРЕКТИРОВКИ

СОВПАДЕНИЯ	41.3%	СОВПАДЕНИЯ	39.93%
ОРИГИНАЛЬНОСТЬ	58.7%	ОРИГИНАЛЬНОСТЬ	60.07%
ЦИТИРОВАНИЯ	0%	ЦИТИРОВАНИЯ	0%
САМОЦИТИРОВАНИЯ	0%	САМОЦИТИРОВАНИЯ	0%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 12.06.2024

ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 12.06.2024 17:14

Структура
документа:

Проверенные разделы: титульный лист с.1, основная часть с.2-65

Модули поиска:

Шаблонные фразы; IEEE; Цитирование; Диссертации НББ, СМИ России и СНГ; Библиография; Патенты СССР, РФ, СНГ; Коллекция НБУ; Переводные заимствования*; СПС ГАРАНТ; аналитика; Публикации eLIBRARY; ИПС Адилет; Перефразирования по коллекции IEEE; Переводные заимствования IEEE; Переводные заимствования по Интернету (EnRu); Перефразирования по СПС ГАРАНТ; аналитика; Медицина; Кольцо вузов; Перефразирования по Интернету (EN); Публикации eLIBRARY (переводы и перефразирования); Перефразирования по Интернету; Публикации РГБ (переводы и перефразирования); Переводные заимствования по коллекции Интернет в английском сегменте; Публикации РГБ; Перефразированные заимствования по коллекции Интернет в английском сегменте; Интернет Плюс*; Перефразированные

Работу проверил: Головяшкин Алексей Николаевич

ФИО проверяющего

Дата подписи:

13.06.24г.



Подпись проверяющего



Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет

Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях

ПРИЛОЖЕНИЕ Д

(обязательное)

Презентация

МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кафедра «Нано- и микроэлектроника»

Система автоматизированного управления газодинамическим стендом

Выполнил студент гр. 22ПДм1
Руководитель к.т.н., доцент каф. НиМЭ

Ганин И.И.
Аношкин Ю.В.

ПЕНЗА 2024

1

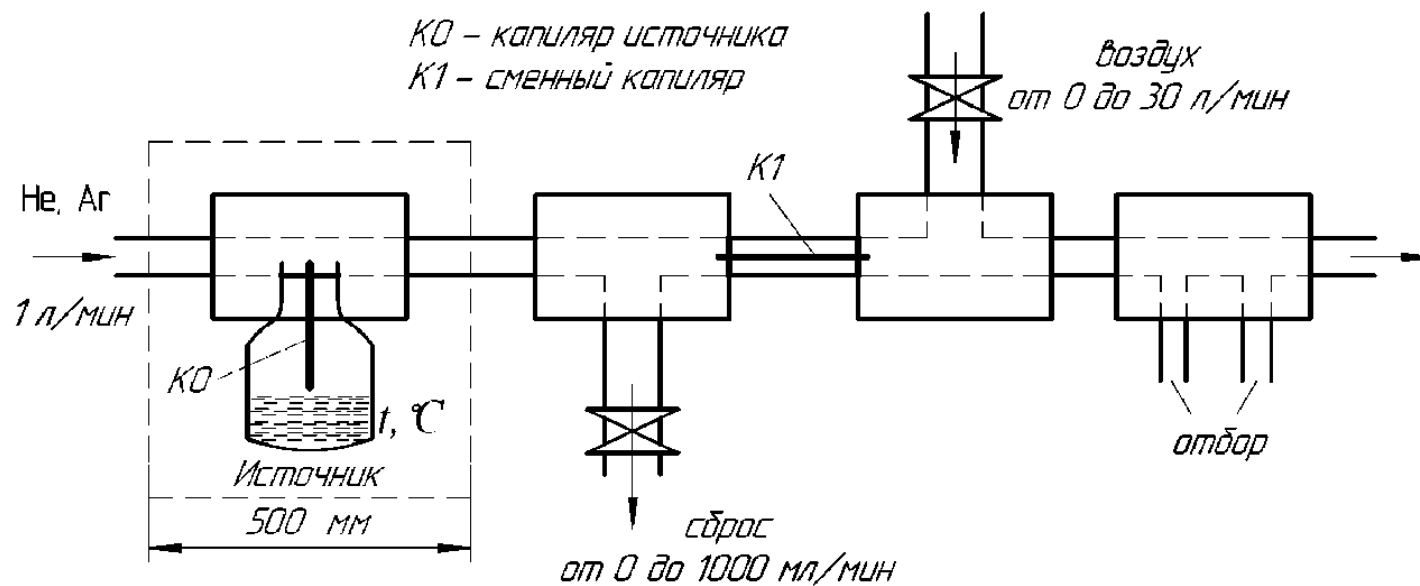
Цель ВКР:

разработать автоматизированную систему управления газодинамическим стендом для создания парогазовых смесей с заданной концентрацией контрольного продукта.

Задачи ВКР:

- выполнить анализ компонентов электроники используемых в системах автоматизации, как для технологических процессов, так и для систем автоматики по безопасности;
- выполнить анализ применяемых измерительных приборов используемых в системе автоматизации;
- выполнить анализ исполнительных компонентов применяемых в качестве движущей силы в системах автоматизации;
- выполнить анализ микропроцессорной техники, которые упрощают систему управления технологическим процессом;
- разработать автоматизированную систему управления газодинамическим стендом

Газодинамический стенд



Структурная схема автоматизированной системы управления

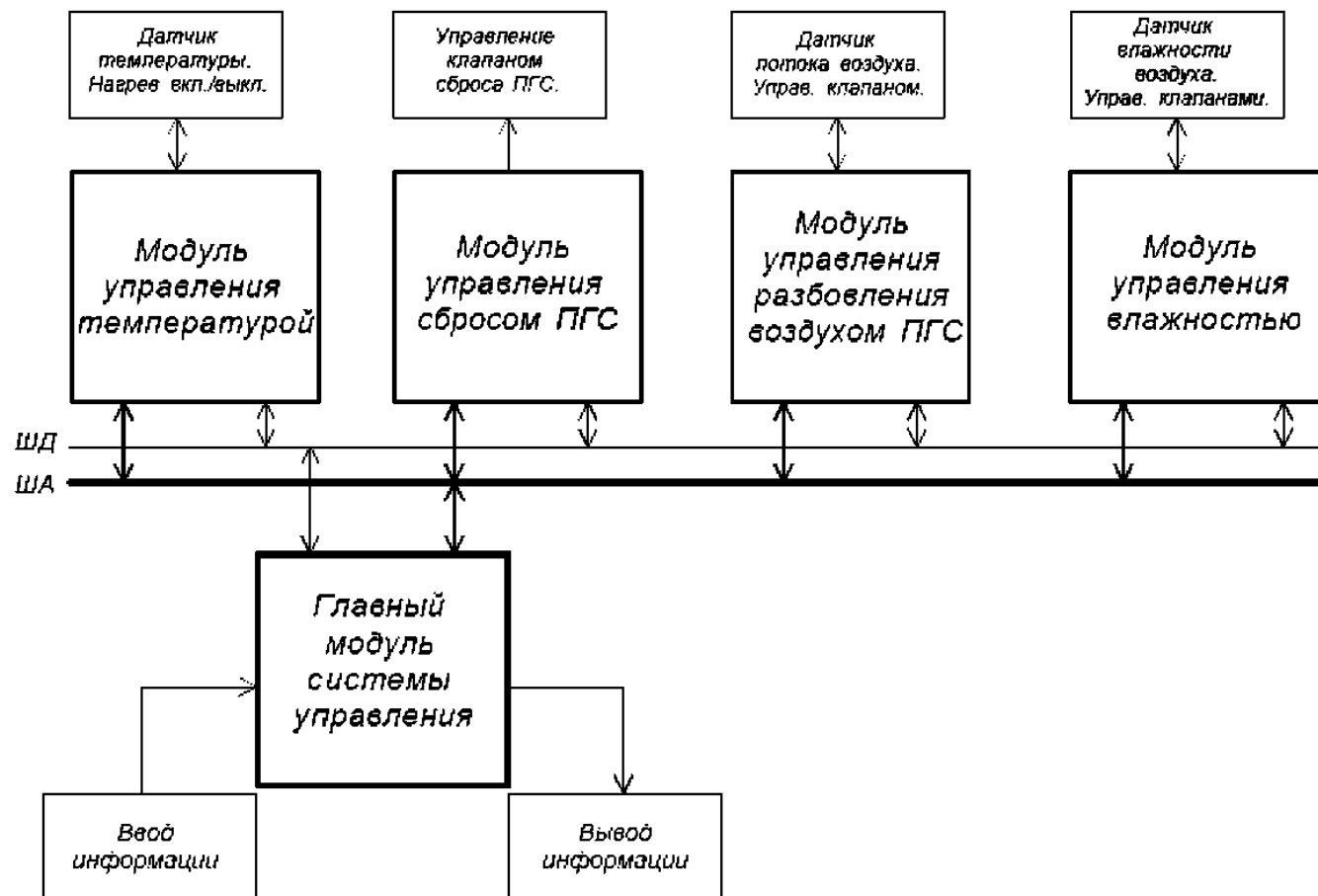
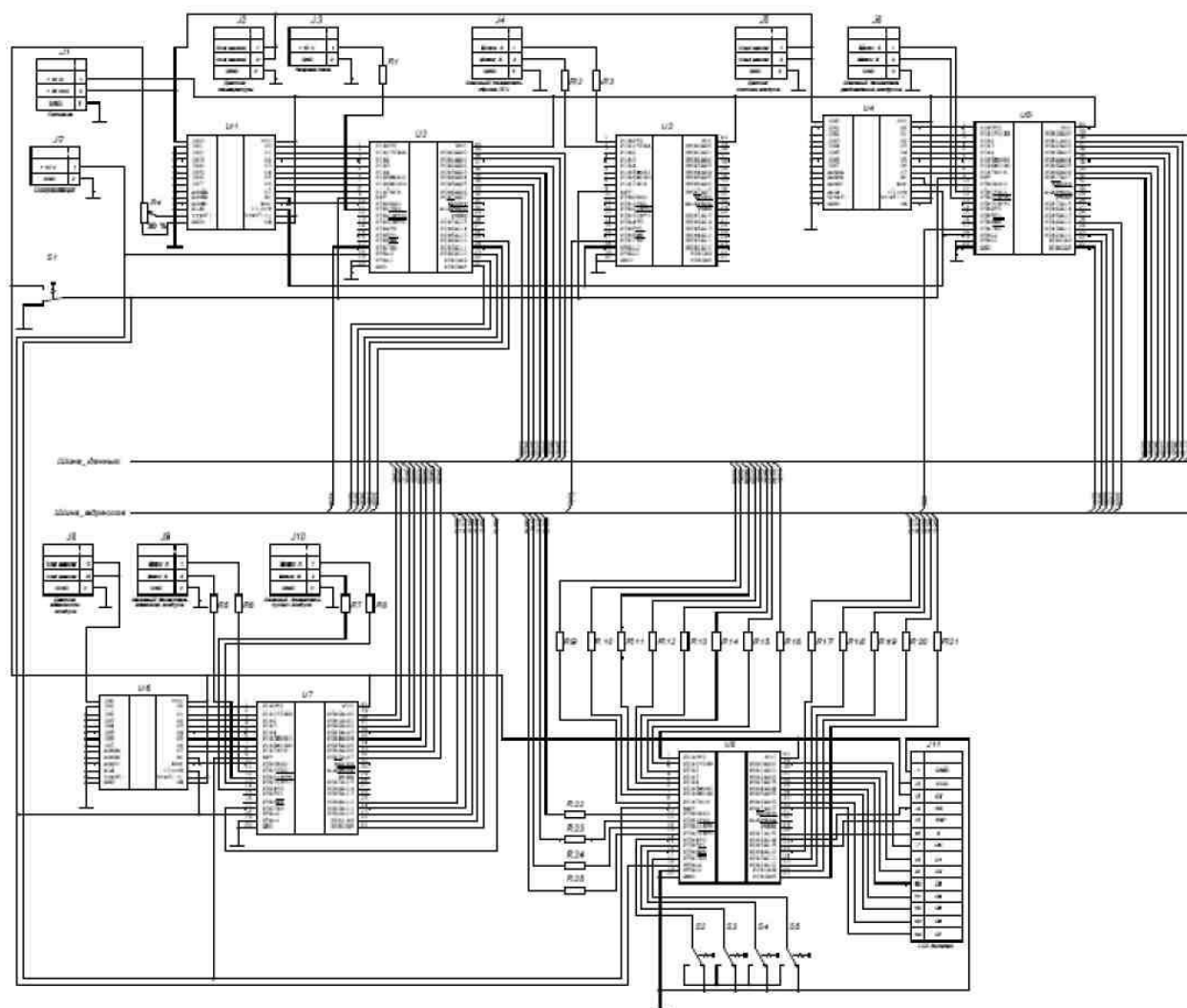
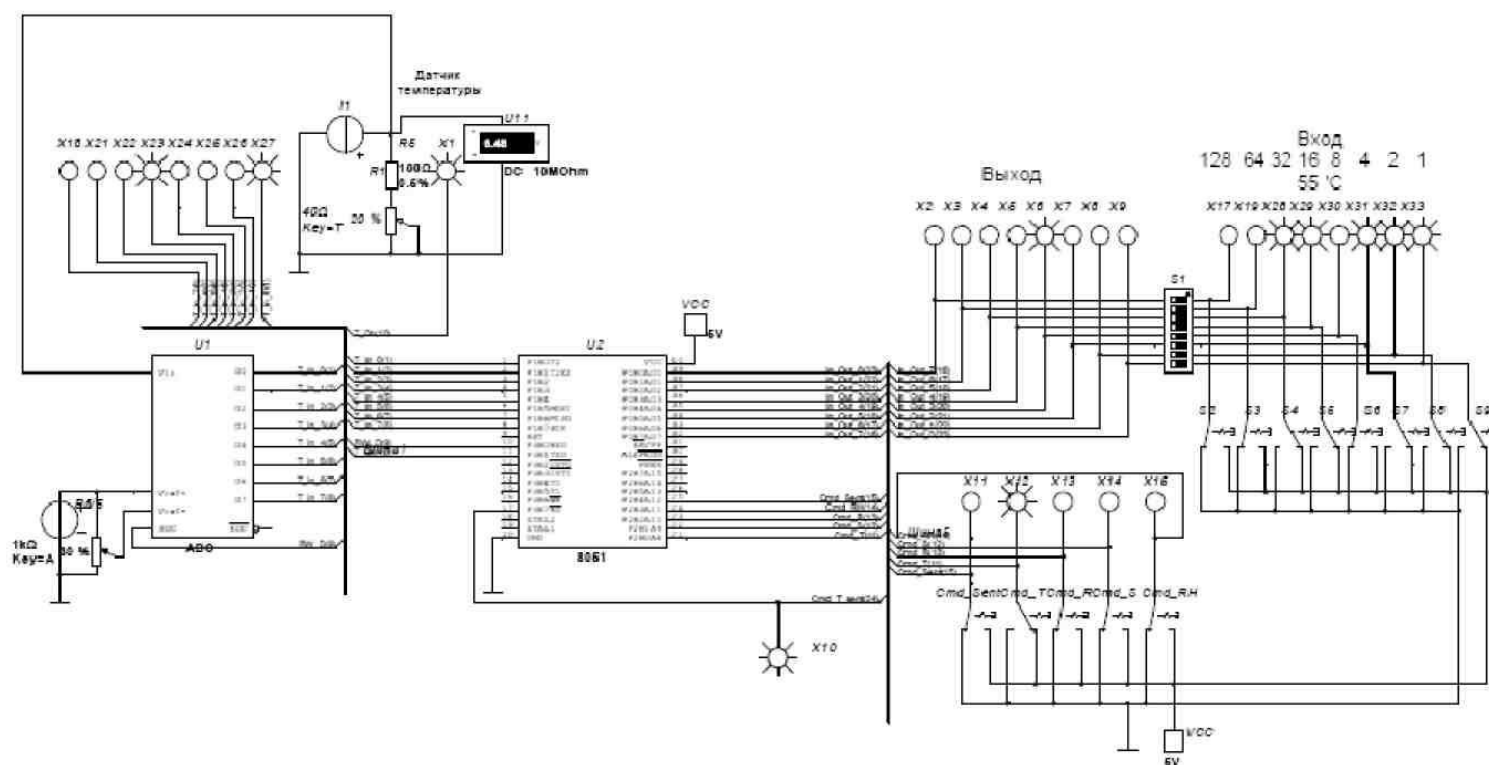


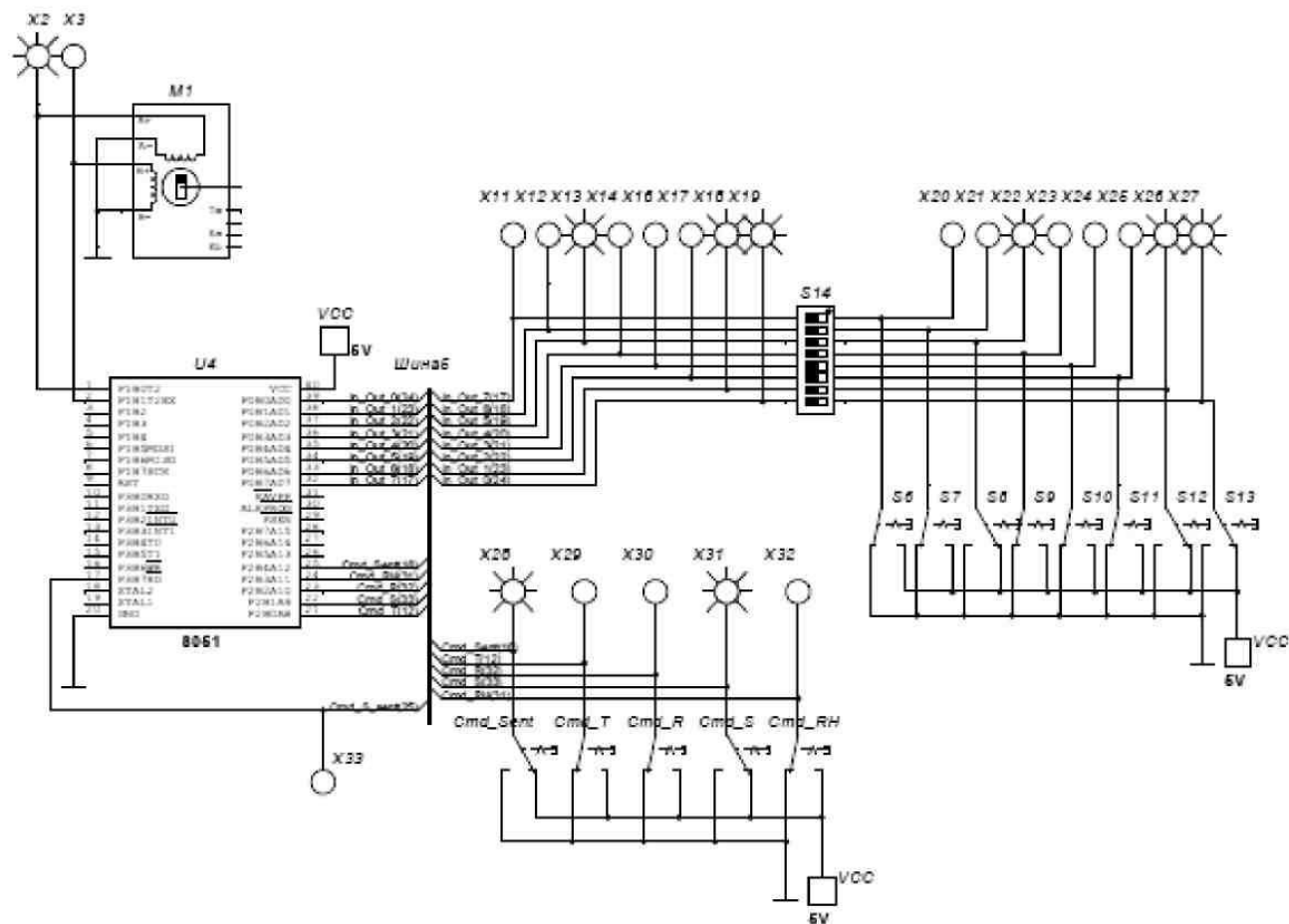
Схема электрическая принципиальная



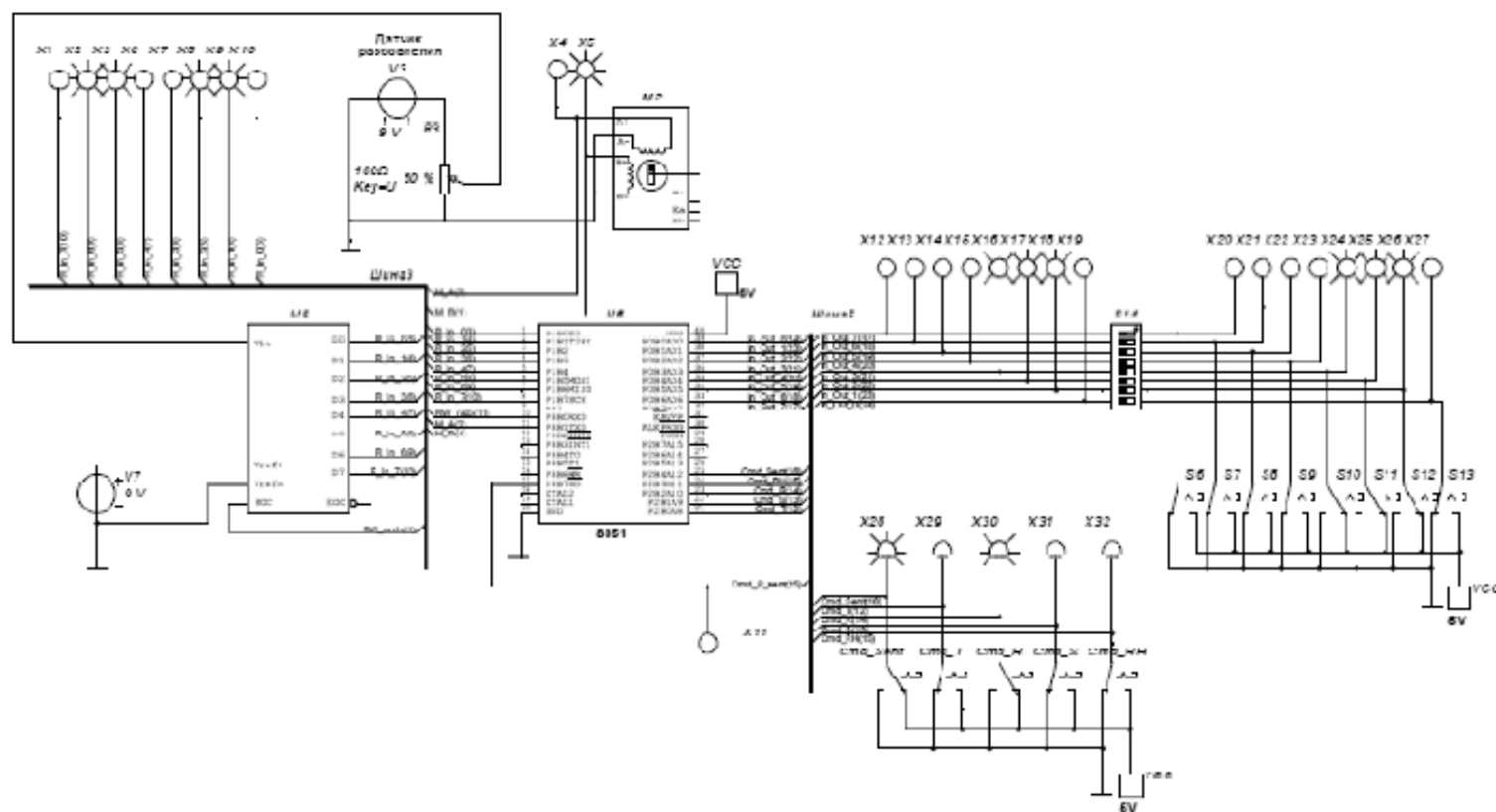
Моделирование работы модуля управления температурой



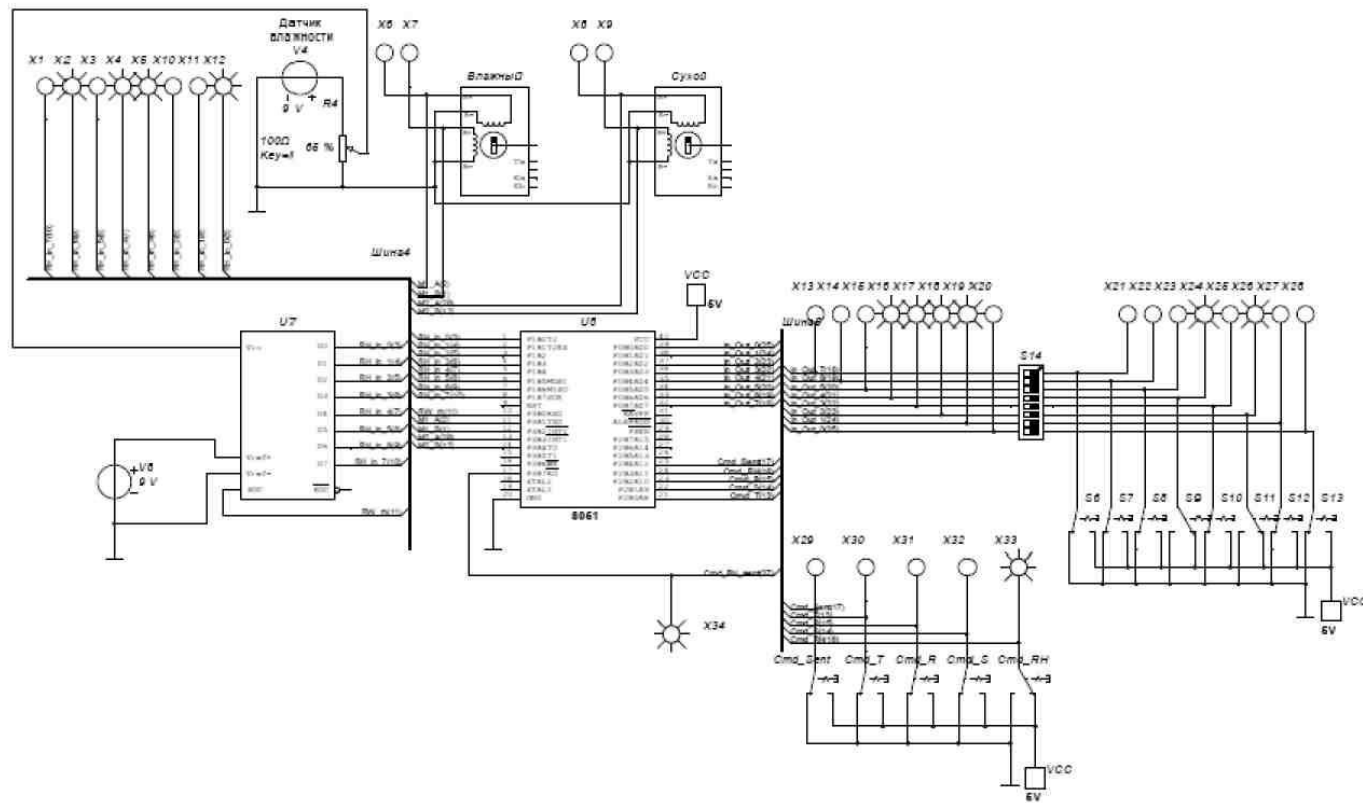
Моделирование работы модуля управления сбросом парогазовой смеси



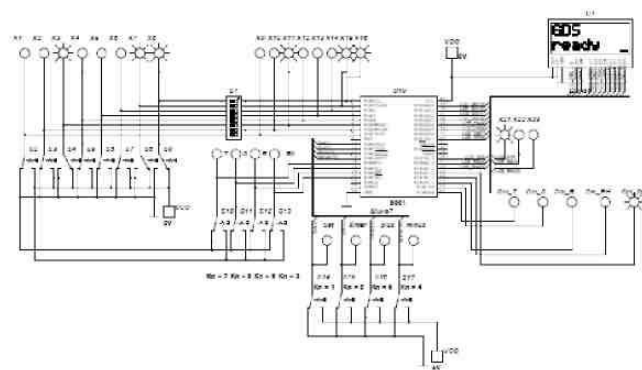
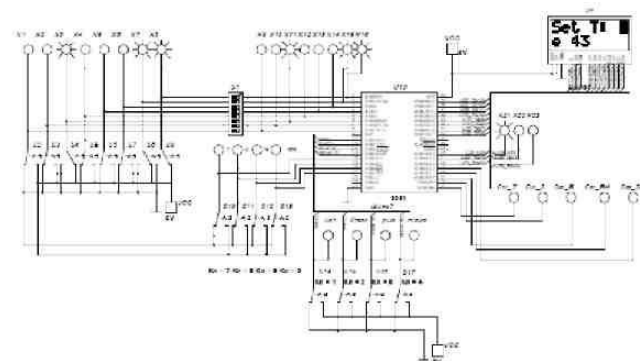
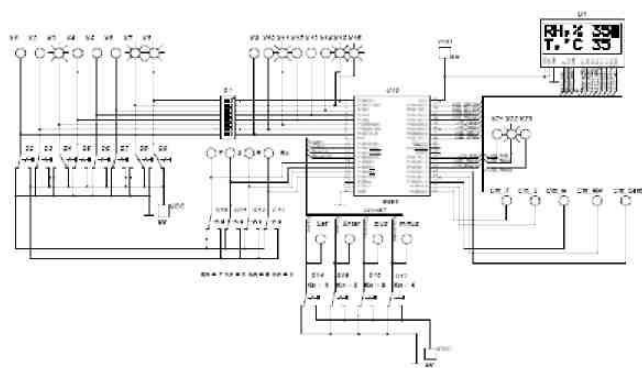
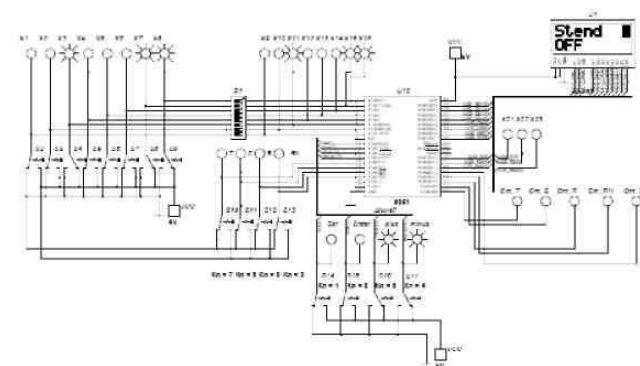
Моделирование работы модуля управления разбавлением воздухом



Моделирование работы модуля управления влажностью воздуха



Моделирование работы главного модуля управления



ВЫВОДЫ по ВКР:

1. Проведен анализ компонентов электроники используемых для проектов автоматизации, как технологических процессов, так и систем автоматики;
2. Проведен анализ измерительных приборов применяемых в системе автоматизации;
3. Проведен анализ исполнительных компонентов применяемых в качестве движущей силы в системах автоматизации;
4. Проведен анализ микропроцессорной техники, которые упрощают систему управления технологическим процессом;
5. Разработана автоматизированная система управления газодинамическим стендом, что позволяет упростить процесс создания парогазовых смесей с заданной концентрацией контрольного вещества.

Спасибо
за
внимание