|  |
| --- |
| **МИНОБРНАУКИ РОССИИ**  федеральное государственное бюджетное образовательное  учреждение высшего образования  **«Национальный исследовательский университет «МЭИ»** |

|  |  |
| --- | --- |
| **Институт** | ИВТИ |
| **Кафедра** | УИТ |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(бакалаврская работа)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Направление** | | | | 27.03.04 Управление в технических системах | | | | | | | | | | |
|  | | | | | (код и наименование) | | | | | | | | | |
| **Образовательная программа** | | | | | | | | | Управление и информатика в технических системах | | | | | |
| **Форма обучения** | | | | | | | | очная | | | | | | |
|  | | | | | | | | (очная/очно-заочная/заочная) | | | | | | |
| **Тема:** | Разработка системы контроля и управления вентиляцией офисного помещения на базе микроконтроллера Arduino | | | | | | | | | | | | | |
| **Студент** | | | А-01-19 | | | | | | |  | | | Изотов А.А. | |
|  | | группа | | | | | | | | подпись | | | фамилия и инициалы | |
| **Руководитель ВКР** | | | | | | |  | | старший преподаватель | | | |  | Баларев Д.А. |
|  | | | | | | | уч. степень | | должность | | | | подпись | фамилия и инициалы |
|  | | | | | | | | | | | | | | |
| организация | | | | | | | | | | | | | | |
| **«Работа допущена к защите»** | | | | | | | | | | | | | | |
| **Заведующий кафедрой** | | | | | д.т.н. | | | | | | доцент |  | | Бобряков А.В. |
|  | | | | | уч. степень | | | | | | звание | подпись | | фамилия и инициалы |
|  | | | | | | | | | | | | **Дата** | |  |
|  | | | | | | **Москва, 2023** | | | | | | | |  |

|  |
| --- |
| **МИНОБРНАУКИ РОССИИ**  федеральное государственное бюджетное образовательное  учреждение высшего образования  **«Национальный исследовательский университет «МЭИ»** |

|  |  |
| --- | --- |
| **Институт** | ИВТИ |
| **Кафедра** | УИТ |

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

**(бакалаврскую работу)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Направление** | | | | 27.03.04 Управление в технических системах | | | | | | | | | |
|  | | | | | (код и наименование) | | | | | | | | |
| **Образовательная программа** | | | | | | | | Управление и информатика в технических системах | | | | | |
| **Форма обучения** | | | | | | | очная | | | | | | |
|  | | | | | | | (очная/очно-заочная/заочная) | | | | | | |
| **Тема:** | Разработка системы контроля и управления вентиляцией офисного помещения на базе микроконтроллера Arduino | | | | | | | | | | | | |
| **Студент** | | | А-01-19 | | | | | |  | | | Изотов А.А. | |
|  | | группа | | | | | | | подпись | | | фамилия и инициалы | |
| **Руководитель ВКР** | | | | | |  | | старший преподаватель | | | |  | Баларев Д.А. |
|  | | | | | | уч. степень | | должность | | | | подпись | фамилия и инициалы |
|  | | | | | | | | | | | | | |
| организация | | | | | | | | | | | | | |
| **Заведующий кафедрой** | | | | | д.т.н. | | | | | доцент |  | | Бобряков А.В. |
|  | | | | | уч. степень | | | | | звание | подпись | | фамилия и инициалы |
| **Место выполнения работы** | | | | | | | | | | ФГБОУ ВО «НИУ «МЭИ» | | | |

**СОДЕРЖАНИЕ РАЗДЕЛОВ ЗАДАНИЯ И ИСХОДНЫЕ ДАННЫЕ**

|  |
| --- |
| На основе микроконтроллера семейства Arduino разработать систему |
| мониторинга температур и автоматизированного управления системой |
| офисной вентиляции офисного помещения. |
| Исходные данные: состав системы вентиляции, модель Arduino и модель |
| сенсорного экрана. |
| Содержание разделов задания: |
| 1. Анализ компонентов системы. |
| 2. Разработка алгоритмов управления и графического интерфейса |
| 3. Разработка схем подключения Arduino к датчикам и исполнительным |
| устройствам |
| 4. Тестирование разработанной системы |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**ПЕРЕЧЕНЬ ГРАФИЧЕСКОГО МАТЕРИАЛА**

|  |  |
| --- | --- |
| **Количество листов** | 5 |
|  |  |
| **Количество слайдов в презентации** | 10 |
|  |  |

**РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА**

|  |
| --- |
|  |
| 1. Изучаем Arduino. Инструменты и методы технического волшебства, Джереми Блум, Издательство BHV, 2015 г, 336 стр., ISBN: 978-5-9775-3585-4 |
| 1. Проекты с использованием контроллера Arduino, Виктор Петин, BHV, 2019 г, 496 стр., ISBN: 978-5-9775-4004-9 |
| 1. Виктор Петин: Издательство BHV; Проекты с использованием контроллера Arduino. [книга], 2021 г. |
| 1. Нимич Г.В., Михайлов В.А., Бондарь Е.С. Издательство: ТОВ Видавничий будинок Аванпост-Прим; Современные системы вентиляции и кондиционирования воздуха [книга], 2003 г. |

***Примечания:***

1. Задание брошюруется вместе с выпускной работой после титульного листа (страницы задания имеют номера 2, 3).
2. Отзыв руководителя, рецензия(и), отчет о проверке на объем заимствований и согласие студента на размещение работы в открытом доступе вкладываются в конверт (файловую папку) под обложкой работы

Аннотация

Выпускная квалификационная работа на тему: «Разработка системы контроля и управления вентиляцией офисного помещения на базе микроконтроллера Arduino».

В работе рассматриваются вопросы подбора компонентов системы контроля и управления вентиляцией. Разработка алгоритма управления компонентами системы, а также интерфейса для взаимодействия с системой. Программная реализация разработанного алгоритма. Физическая реализация.

В выпускной квалификационной работе представлены: введение, обзор платформы и компонентов системы, обзор исполнительных устройства, алгоритмы управления.

Список сокращений

ПЗУ – постоянное запоминающее устройство

USB – universal serial bus (универсальная последовательная шина)

ОЗУ – оперативное запоминающее устройство

UART – universal asynchronous receiver-transmitte (универсальный асинхронный приёмопередатчик)

TFT – thin film transistor (тонкоплёночный транзистор)

ТЭН – трубочный электронагреватель

ПИД–пропорционально-интегрально-дифференцирующий регулятор

ПИН– personal identification number (персональный идентификационный номер)

Содержание

[ВВЕДЕНИЕ 9](#_Toc138019686)

[Глава 1. Анализ и обзор компонентов, общей вид системы. 11](#_Toc138019687)

[1.1. Контроллер и его компоненты 11](#_Toc138019688)

[1.1.1. Микроконтроллер Arduino Mega 11](#_Toc138019689)

[1.1.2. Дисплей ILI9481 14](#_Toc138019690)

[1.1.3. Датчик температуры и влажности DHT-22 14](#_Toc138019691)

[1.1.4. Датчик DS18B20 16](#_Toc138019692)

[1.2. Исполнительные устройства 17](#_Toc138019693)

[1.2.1. Обзор канальных нагревателей и принципов управления ими 17](#_Toc138019694)

[1.2.2. Выбор канального нагревателя 22](#_Toc138019695)

[1.2.3. Двигатель канального вентилятора 24](#_Toc138019696)

[1.2.4. Электропривод жалюзи 26](#_Toc138019697)

[1.2.5. Автотрансформатор 27](#_Toc138019698)

[1.3. Фильтры 29](#_Toc138019699)

[1.4. Подключение компонентов контроллера 30](#_Toc138019700)

[1.5. Подключение исполнительных устройств 31](#_Toc138019701)

[1.6. Общий вид системы 31](#_Toc138019702)

[Вывод по главе 1 32](#_Toc138019703)

[Глава 2. Разработка системы, алгоритмов управления и интерфейса 33](#_Toc138019704)

[2.1. Алгоритм управления для поддержки температуры 33](#_Toc138019705)

[2.2. Интерфейс системы 34](#_Toc138019706)

[2.3. Алгоритм программной реализации микроконтроллера 38](#_Toc138019707)

[Вывод по главе 2 44](#_Toc138019708)

[Глава 3. Схема подключения оборудования 46](#_Toc138019709)

[3.1 Блок реле 46](#_Toc138019710)

[3.2 Подключение дисплея 47](#_Toc138019711)

[3.3 Схема подключения датчиков 49](#_Toc138019712)

[3.4 Подключение исполнительных устройств 49](#_Toc138019713)

[Вывод по главе 3 50](#_Toc138019714)

[Глава 4. Первоначальное тестирование 51](#_Toc138019715)

[4.1 Общая схема тестирования 51](#_Toc138019716)

[4.2 Тестирование датчиков 53](#_Toc138019719)

[4.3 Тестирование экрана 53](#_Toc138019721)

[Вывод по 4 главе 53](#_Toc138019723)

[ЗАКЛЮЧЕНИЕ 54](#_Toc138019724)

[СПИСОК ЛИТЕРАТУРЫ 55](#_Toc138019725)

ВВЕДЕНИЕ

В современных офисных помещениях комфортные условия работы играют важную роль для повышения производительности и благополучия сотрудников. Одним из ключевых аспектов комфорта является качество воздуха, которое обеспечивается эффективной системой вентиляции. Однако традиционные системы вентиляции могут быть ограничены в своих возможностях управления и контроля.

В последние годы развитие микроконтроллеров, таких как Arduino, открыло новые возможности для разработки систем контроля и управления, включая системы вентиляции. Arduino предоставляет гибкую и доступную платформу для создания интеллектуальных систем, позволяющих контролировать и регулировать работу вентиляционных систем с учетом различных факторов.

Целью данной работы является разработка системы контроля и управления вентиляцией офисного помещения на базе микроконтроллера Arduino. В работе будут рассмотрены основные компоненты и модули, необходимые для реализации такой системы, включая датчики для измерения параметров воздуха, активаторы для управления вентиляторами и клапанами, а также программное обеспечение для управления, обработки данных и представления данных.

При разработке системы будут учтены различные факторы, влияющие на качество воздуха в офисном помещении, такие как температура и влажность. Пользователи сами будет задавать необходимое значение температуры, будут разработаны алгоритмы управления, позволяющие автоматически регулировать работу вентиляционной системы для поддержания комфортных условий.

Такая система контроля и управления на базе микроконтроллера Arduino имеет ряд преимуществ, включая низкую стоимость, гибкость в настройке и возможность интеграции с другими устройствами и системами управления в офисе. Она позволяет достичь более эффективного использования ресурсов и повысить качество воздуха, что способствует улучшению рабочей среды и благополучию сотрудников.

В работе будут представлены результаты разработки и тестирования системы контроля и управления вентиляцией на базе микроконтроллера Arduino.

1. Анализ и обзор компонентов, общей вид системы.
2. Контроллер и его компоненты
3. Микроконтроллер Arduino Mega

«Arduino» — это общее название платформы для разработки автоматических систем различного уровня сложности. «Arduino» имеет огромный модельный ряд, включающий в себя: «Arduino Nano», «Ardiuno UNO», «Ardiuno Mega». На основе этой платформы можно разработать различные системы, такие, как системы поддержания климата, «умный» дом, полив растений, различные системы безопасности, парктроники и многое другое. В основе платформы лежит микроконтроллер, при помощи которого можно выдавать и измерять напряжения, производить вычисления, запоминать данные. Как следствие можно считывать данные с датчиков (как аналоговых, так и с дискретных), принимать данные по различным сетям (например Wi-Fi или мобильная сеть), чтения с ПЗУ, управлять различными устройствами или системами.

Также, помимо котроллера, на плате находится его обвязка, которая включает в себя: программатор для загрузки прошивки со входом USB, стабилизатор напряжения 5В для питания микроконтроллера, различные входы и выходы для подключения (далее порты). Все модели имеют одинаковый функционал и отличаются только размером, микроконтроллерами, количеством ОЗУ и ПЗУ, а также количеством портов. Порты в «Arduino» имеются как аналоговые, так и цифровые, которые могут принимать и выдавать цифровой сигнал (0 и 5 В), аналоговые порты могут ещё и принимать аналоговый сигнал от 0 до 5В с шагом 5мВ.

В проекте используется «Arduino Mega», содержащая 54 дискретных порта, 16 аналоговых, 4 UART порта, 16 МГц микроконтроллер ATmega2560, порт USB, порт питания power jack, кнопку перезагрузки. Напряжение для операций 5В, рекомендуемое напряжение питания 7-12В, максимальное входное напряжение питания 20В, максимальная сила тока для входных и выходных портов 20мА, максимальный ток для порта 3.3В 50мА, Флеш-память 256 Кбайт (8 Кбайт заняты загрузчиком), ОЗУ 4 Кбайт. Размеры за исключением толщены 100 на 50 мм.

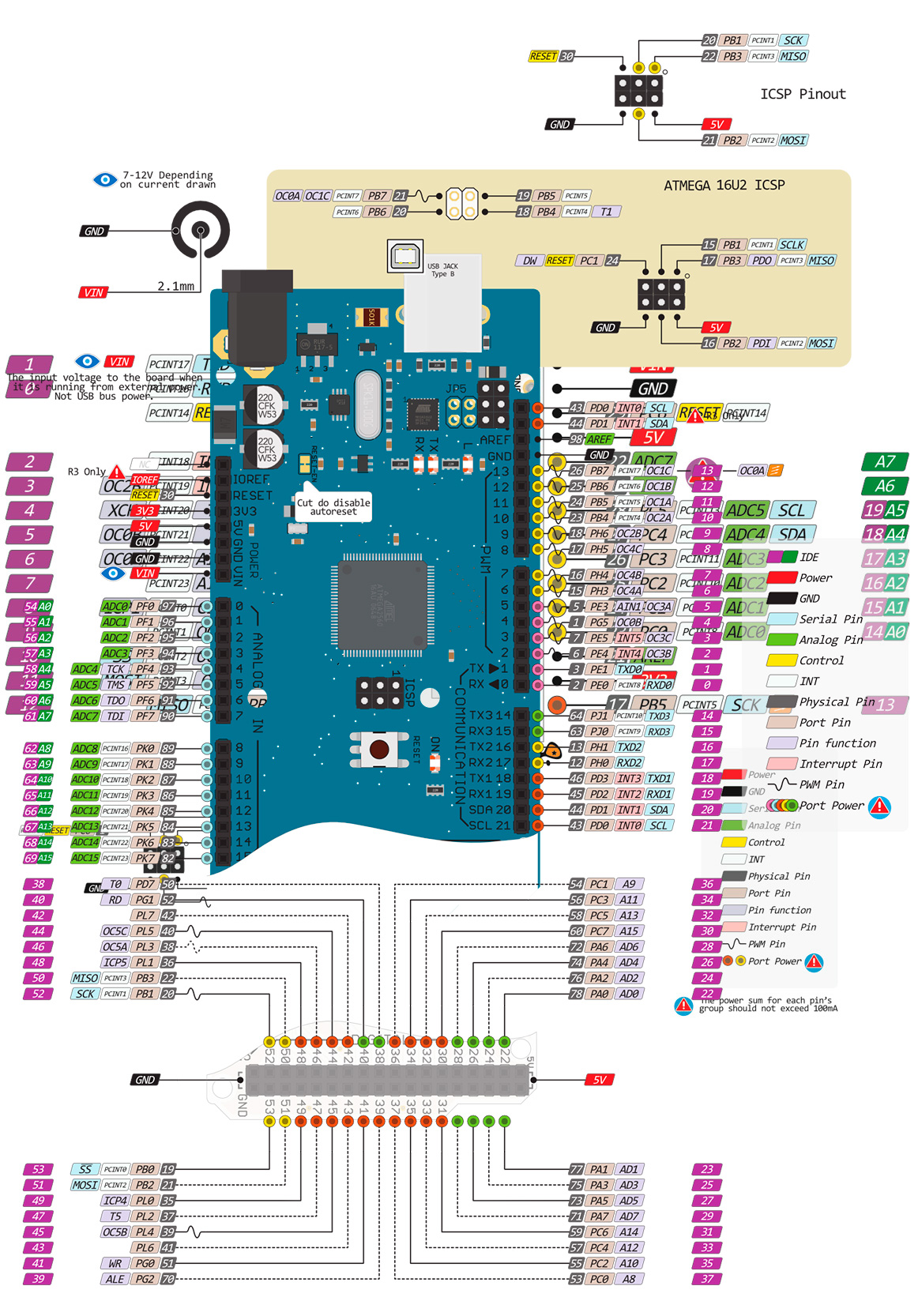


Рис. 1 схема контактной распиновки Arduino MEGA2560

1. Дисплей ILI9481

Цветной сенсорный TFT-дисплей ILI9481-экран разработанный для Arduino и имеет соответствующие библиотеки для работы с ним. Разрешение экрана 480x320 пикселей, а размер экрана 3.5 дюйма, из-за чего он перекрывает большое количество портов Arduino Mega, что можно увидеть на Рис. 2, его размеры на 55 мм за исключением толщены.

На данном дисплее будет разработан интерфейс, он поддерживает цветную отрисовку и сенсорное резистивное управление, т.е. при нажатии мы можем фиксировать напряжение, выдаваемое по вертикали и горизонтали.

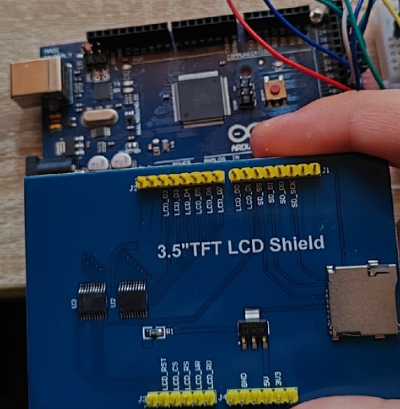


Рис. 2 Дисплей и Arduino

1. Датчик температуры и влажности DHT-22

Датчик температуры и влажности DHT-22-цифровой реализованный на микросхеме AM2302. Он используется для измерения влажности и первоначального быстрого замера температуры при инициализации устройства. Он не используется для замера температуры в штатной работе системы ввиду его относительно низкой точности (0,5 градусов). Технические характеристики:

Напряжение питания: 3 ... 6В

Диапазон измерения влажности: 0 ... 100%

Диапазон измерения температуры: -40℃...+80℃

Погрешность измерений влажности: 2%

Погрешность измерений температуры: 0.5°С

Шаг измерения влажности: 0.1%

Шаг измерения температуры: 0.1 °С

Период измерений: 1 сек.

Интерфейс: onewire, 1-проводной

Шаг выводов: 2,54 мм

Размеры: 15 x 25 x 7.7 мм

Вес: 2.2 гр.

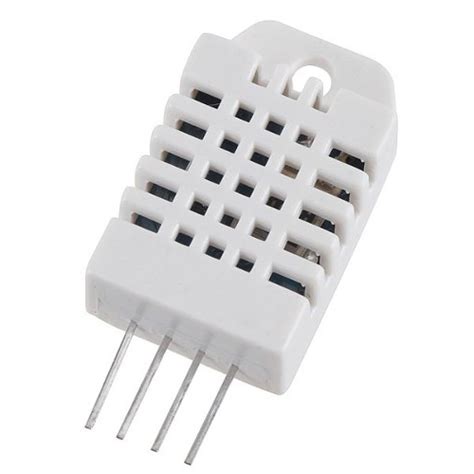


Рис. 3 Датчик DHT-22

1. Датчик DS18B20

DS18B20 — это цифровой датчик, который применяется для измерения температуры (диапазон -55 °C до 125 °C), а также обладает программируемой уникальной точностью (до 12 бит).

Один из ключевых элементов в структуре термодатчика следует отнести «64-BITROM AND 1-WIREPORT». Это модуль структуры, отвечающий за хранение уникального кода устройства и передачу этого кода во внутреннюю память DS18B20 «SCRATCHPAD». «SCRATCHPAD», в свою очередь, взаимодействуя с регистрами «MEMORY CONTROL LOGIC» и «1-Wire», подает сигнал связи следующим принципиально-важным блокам датчика:

«TEMPERATURE SENSOR» (система, предназначенная для считывания преобразованных показателей температур);

«CONFIGURATON REGISTER» (структура, отвечающая за настройку уникальной программируемой точности, которая варьируется от 9 бит до 12 или, если рассматривать градусы Цельсия, от 0.5 °C до 0.0625 °C.);

«8-BIT CRC GENERATOR» (система, предназначенная сугубо для защитной функции);

«ALARM HIGH TRIGGER» (блок, ограничивающий нижние пределы температуры DS18B20);

«ALARM LOW TRIGGER» (система, ограничивающая верхние пределы температуры DS18B20).

Таким образом можно создавать кластеры из этих датчиков и замерять температуру в разных концах помещения либо же в разных помещениях и масштабировать систему.



Рис. 4 Датчик DS18B20

1. Исполнительные устройства

Исполнительные устройства необходимые для работы системы включают в себя: привод жалюзи, автотрансформатор, двигатель вентилятора и канальный нагреватель. Касательно привода жалюзи и двигателя вентилятора, их можно брать любыми в зависимости от объёма воздуха, который необходимо прокачивать, это решается на основании объёма помещения и количества люде находящихся в нём, так кратность воздухообмена можно найти при помощи отношения расхода воздуха в час к объёму помещения, также необходимо учитывать, что количество приточного воздуха равно количеству вытяжного. Система рассчитана на пятиступенчатый автотрансформатор, то есть можно устанавливать 5 режимов работы вентилятора, которые настраиваются под нужды объекта. Выбор же калорифера, тема для отдельного исследования, так как необходимо понимать целесообразность того или иного вида калориферов.

1. Обзор канальных нагревателей и принципов управления ими

Канальный нагреватель, он же калорифер, при наличии вентилятора – прибор в системах вентиляции и воздушного отопления для нагревания приточного воздуха. Прибор состоит из труб, ТЭНов или других нагревательных элементов, по которым циркулирует электрический ток, вода, пар, масло или другие жидкостные, или газовые теплоносители. По принципу передачи тепловой энергии можно различать: электрические, паровые и водяные (могут быть подключены к системе центрального отопления) канальные нагреватели. В зависимости от конфигурации сечения вентиляции могут быть использоваться воздухонагреватели различных форм: прямоугольной и круговой.

Существуют канальные нагреватели с утилизацией тепла (рекуперация). В этом случае холодный приточный воздух нагревается за счет теплообмена с удаляемым теплым воздухом. Воздушные потоки при этом не смешиваются.

Так как устройства нагревают приточный воздух, то канальные и водяные воздухонагреватели можно эксплуатировать даже для нагрева взрывоопасной газовой среды. Существуют различные модификации калориферов:

• Водяного типа: КСк, КСн, КВС, КВБ

• Парового: КП/КПСк, КФБ, КФС, КП-Сн, КПС, КПЮ

• Электрического: СФО, ЭК

• КФБО, КФСО

Сами по себе калориферы очень надёжные устройства, не нуждающиеся в частом обслуживании, качество же их работы зависит от их системы автоматики. Также необходимо смотреть на мощность калорифера, для прогрева необходимого объёма воздуха в помещении в пределах нужного времени.

Работа приточной вентиляции: наружный воздух, поступает через воздухозаборную решётку и проходя через жалюзийную решётку, попадает в место с фильтрами, где должна происходить очистка воздуха от пыли и прочих примесей. Далее очищенный воздух попадает в калорифер, в котором происходит нагрев. После чего нагретый воздух за счёт вентиляторов попадает в приточный канал или сразу же в помещение через жалюзи (также можно устанавливать жалюзи до системы фильтров). На Рис. 5 представлена работа вентиляции на примере водяного калорифера.

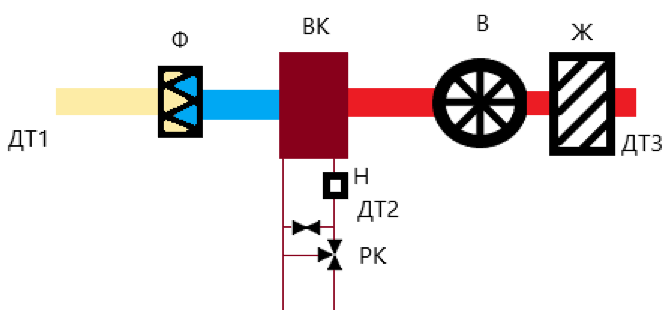


Рис. 5 Общий принцип работы системы вентиляции

• ДТ1 – датчик температуры заборного воздуха

• ДТ2 – датчик температуры воды

• ДТ3 – датчик температуры помещения (в канале)

• Ф – фильтры

• ВК – водяной калорифер

• Н – циркулирующий насос

• РК – регулирующий клапан

• В – вентилятор

• Ж – жалюзи

Рассмотри подробно водяной калорифер:

Регулирующая арматура водяного калорифера (обвязка) в зависимости от источника горячей воды представляется двумя способами:

a) Потребление воды из городской сети, где нет необходимости фиксировать объёмы обратной воды, но появляется критическая необходимость регулировать температуру обратной воды, для этого применяется двух.

b) Потребление из местной бойлерной или котельной, где необходимо учитывать расход обратной воды и применяется только трёхходовой вентиль.

Сами же системы по принципу работы не отличаются. Главное отличие в том, что в первом случае, возможно полное перекрытие потока в обратной магистрали.

Функции системы автоматики:

1. Включение/выключение системы (вручную, по таймеру, в аварийных случаях).

2. Поддержка требуемой температуру в рабочем режиме.

3. Защита калорифера от разморозки.

4. Поддержка температуры обратной воды.

5. Тренинговый старт насоса.

Система автоматики должна включать в себя несколько режимов:

1. предварительный прогрев

2. запуск

3. работа

4. дежурный режим

Для хорошей управляемости и динамичности водяной контур обвязки калорифера должен быть как можно короче, также для защиты от замерзания устанавливается термостат защиты от заморозки, который прикрепляется к теплообменной поверхности калорифера и срабатывает при сильном снижении температур или переохлаждении калорифера.

Основные исполнительные механизмы системы: привод жалюзийной решётки, контактор вентилятора, пускатель насоса и регулируемый клапан. Привод жалюзийной решётки и контактор вентилятора можно объединить в единую группу, таким образом при подаче сигнала будет открываться решётка и включаться вентилятор.

В периоды отрицательных температур наружного воздуха перед пуском системы будет требоваться предстартовый прогрев. Когда система находится в дежурном режиме (ещё не запущена) необходимо поддерживать воду в обратной магистрали. Для поддержки этой функции клапан почти закрыт и открытие дроссельной заслонки со включением вентилятора грозит разморозкой калорифера, таким образом во время прогрева необходимо следить за температурой обратной воды во избежание резкого падения температуры подачи воздуха. Прогрев также необходим чтобы в помещение или воздуховод подавался нагретый воздух. Прогрев может осуществляться как по времени, так и по достижении определенной температуры обратной воды, однако наиболее оптимальным является прогрев воды до конкретных значений температуры, по заданным интервалам времени.

После прогрева необходимо выйти на режим, при этом отдавать наибольший приоритет температуре обратной воды, ведь из-за холодного воздуха в зимние периоды температура воды может резко снижаться. В разные моменты времени система обращает внимание на показания различных датчиков, а именно: если температура обратной воды в норме, то система работает для поддержания температуры помещения, как только обратная вода выходит из необходимой температурной зоны система работает для возвращения температуры обратной воды к приемлемому значению.

Как только температура обратной воды и помещения (воздуховода) достигла нужного нам значения система выходит на рабочий режим. При выключении приточной вентиляции система переходит в дежурный режим.

Для данной системы недопустимо использование on/off регулирования, а именно мы не можем постоянно полностью открывать и закрывать клапаны входа и выхода воды в системе, во избежание отправки в отопительную систему выбросов холодной воды. Таким образом появляется необходимость использования ПИД или нейросетевого регулятора, что увеличит сложность системы.

Электрический калорифер, в отличие от паровых и водяных установок, не предполагают подведения дополнительных коммуникаций, что позволяет легко монтировать их в вентиляционный ходы или же обслуживать малогабаритные помещения. Управление ими может быть организованно простейшим on/off регулированием. Одним из главных недостатков можно выделить выжигания им воздуха, и существенный расход электроэнергии. Также они обладают маленькой инерционностью в отличии от водяных калориферов, из-за чего быстро охлаждаются. Они не зависят от наружной температуры, благодаря чему можно обойтись лишь одним датчиком температуры помещения и учитывать только его.

1. Выбор канального нагревателя

Электрический калорифер целесообразно брать для отопления небольших помещений или для домашнего отопления, ввиду его простоты использования (допускается on/off регулирование, нет необходимости подводки теплоносителя, нужно лишь электричество), и куда более низкой цене. Из недостатков можно выделить цену работы такой системы в зимнее время, так как забор воды из центральной отопительной системы сильно дешевле электричества. Но также мы лишаемся рисков в виде заморозки труб.

Водяной калорифер целесообразно брать, при необходимости прогрева больших помещений на весьма долгий срок ввиду необходимости подключения к ним труб, что сильно увеличивает стоимость системы. В водяном калорифере необходимо либо обучать нейросетевой регулятор, либо настраивать ПИД, что увеличивает сложность разработки такой системы. Также сложность вызывает постоянный контроль температуры воды в калорифере, в случае заморозки есть шанс разрыва труб, из-за чего появляется необходимость в источнике бесперебойного питания, таким образом имеется риск замерзания воды в трубах.

Для дальнейшей работы был выбран электрический калорифер, ввиду его большей надёжности и простоты. Касательно экономической целесообразности нельзя выделить явно лучшее решение, ввиду повышенных рисков для водяного калорифера, так как он потребует больших первоначальных вложений, и его выход из строя может произойти раньше, чем мы перекроем расходы на эксплуатацию электрического калорифера.

В системе предполагается круглый канал диаметром 160мм, в связи с этим можно взять канальный нагреватель «НККм-160-2,0-1ф Ванвент». Характеристики:

* Диаметр 160 мм
* Вес 4,5 кг
* Мощность 2000 Вт
* Количество фаз 1
* Напряжение 230 В

Он был выбран за счёт своей недорогой стоимости и относительно малой мощности 2000 Вт, что позволит управлять им почти любыми реле.



Рис. 6 НККм-160-2,0-1ф Ванвент

1. Двигатель канального вентилятора

В качестве двигателе с вентилятором был выбран «TD500/160 T Mixvent Soler&Palau вентилятор канальный». Вентилятор предназначен для 160 каналов и обладает тремя режимами скоростей, однако с использованием автотрансформатор, мы будем работать только с максимальным режимом работы скорости вентилятора. Также он имеет достоинства центробежных вентиляторов -большое давление на выходе. А также осевых: небольшой уровень шума, малые размеры и большой расход воздуха. Двигатель однофазный, асинхронный с постоянным конденсатором и внешним ротором. Имеет защиту от перегрева ( самовозвратный термопредохранитель). Класс защиты IP 44. Характеристики:

* Диаметр 160 мм
* Уровень шума 33 dB
* Вес 2.7 кг
* Мощность 50 Вт
* Максимальные обороты вентилятора 2500 об/мин
* Напряжение 230 В
* Производительность 580 м3/ч
* Давление 295 Па
* Рабочая температура от -20 до +60 градусов цельсия
* Сила тока 0.22 А



Рис. 7 Канальный двигатель с вентилятором

1. Электропривод жалюзи

В качестве электропривода для жалюзи наиболее удобно использовать с возвратной пружиной, например «341-024D-03 Gruner электропривод с моментом вращения 3 Нм с возвратной пружиной». Электропривод специально разработан для использования с воздушными заслонками и вентилями в системах вентиляции, имеет кнопку ручного управления, механический ограничитель угла поворота, кнопку установки направления вращения. Характеристики:

* Вес 1 кг
* Класс защиты IP 54
* Тип регулирование 2-позиционное
* Возвратная пружина 3 Hm
* Напряжение 24 В



Рис. 8 электропривод с возвратной пружиной

1. Автотрансформатор

Автотрансформатор выбран пятиступенчатый трехфазный «Shuft ATRE-1,5», он предназначен для регулирования скорости вращения и однофазных электродвигателей вентиляторов и расхода воздуха, создаваемого данными вентиляторами. Принцип работы основан на коммутировании выходных клемм.



Рис. 9 пятиступенчатый автотрансформатор

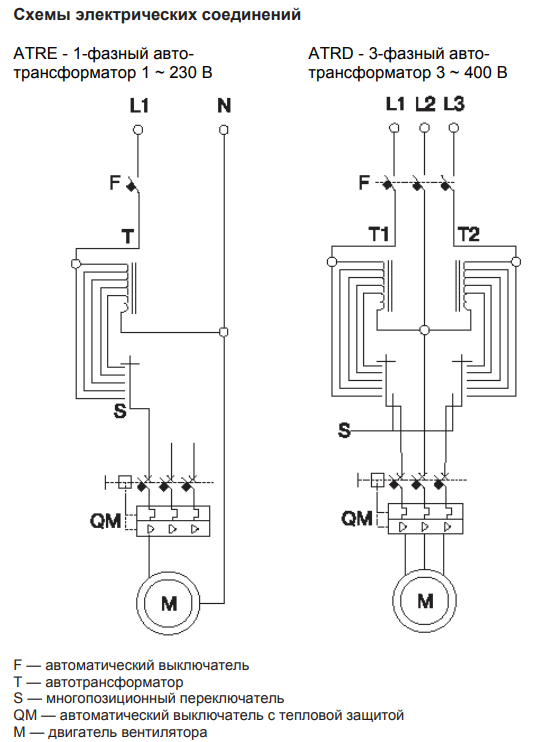


Рис. 10 Схема электрических соединений автотрансформатора

1. Фильтры

В качестве фильтра для круглых воздушных каналов ФЛК, было решено взять «ФЛК 160 Airone». Воздушный фильтр предназначен для очистки воздуха от пыли в системах приточной и вытяжной вентиляции. Фильтр представляет из себя прямоугольный контейнер, в котором находится очиститетельный материал в форме клина. В данном фильтре используется в качестве фильтрующих вставок синтетический материал с классом очистки G3(EU3). Его необходимо менять примерно один раз в месяц.

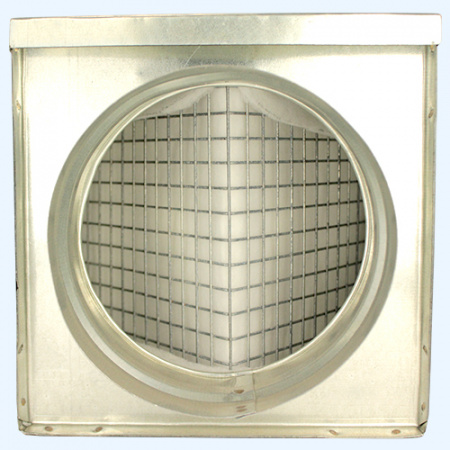


Рис. 11 ФЛК

1. Подключение компонентов контроллера

Компонентами контроллера являются датчики температуры и влажности DHT-22 и температуры DS18B20, сенсорный дисплей ILI9481, светодиоды и резисторы. Присутствуют 7 светодиодов (С1, C2, C3, С4, С5, С6, С7). С1- мигающий светодиод, сигнализирующий нам о том, что система работает, период его мигания 1,5 сек. С2-светодиод, сигнализирующий превышение температуры уставки плюс гистерезис, и необходимости включения вентиляции, без обогрева, он будет гореть пока температура вновь не вернётся к установленной. С3-светодиод, сигнализирует нам о том, что температура ниже температуры уставки минус 0.1 градус (гистерезис системы в силу её инертности было принято взять 0.1 градус Цельсия), и необходимо включить обогрев, он будет гореть пока температура вновь не вернётся к установленной. С4, С5, С6, С7- светодиоды, показывающие режим работы двигателя 0, 1, 2, 3 соответственно.

1. Подключение исполнительных устройств

Исполнительные устройства: электрический канальный нагреватель, привод жалюзи подключены к сети 220 В через нормально открытые реле 1, таким образом, когда появляется сигнал на 43, 45 или 47 пине запускается привод жалюзи для открытия канала. Двигатель с вентилятором связан с автотрансформатором, который в зависимости от поданного ему сигнала выдаёт различное напряжение, то есть режимы работы и запускает двигатель, причём режимы работы завязаны на тех же пинах, что и привод жалюзи, поэтому при запуске двигателя канал будет открыт. Электрический канальный нагреватель в свою очередь связан с 1 нормально открытым реле через 2 нормально открытое реле, которое будет открыто в случае подачи сигнала с 37 пина, который активируется, если температура ниже температуры уставки минус гистерезис. Таким образом для запуска нагревателя необходимо, чтобы канал был открыт (соответственно работает двигатель вентилятора) и был сигнал с 37 пина.

1. Общий вид системы

Целиком система помимо контроллера и исполнительных устройств будет включать в себя другие компоненты вентиляции. К приводу жалюзи необходима сама воздухозаборная решётка, а также необходимы соединительные элементы, по которым будет идти заборный воздух. После того, как воздух пройдет через воздухозаборную решётку и жалюзи он попадает в отсек с фильтрами очистки воздуха, они могут быть как тканевые, так и угольные, затем в отсек с двигателем и вентилятором, после чего идут в глушитель, представляющий собой увеличенный в диаметре соединительный отсек, который глушит звуки работы системы для комфортного пребывания в помещении (шум отталкивается от стенок и тем самым гасится), после чего воздух проходит через канальный нагреватель. В конце под потолком воздух выходит в разветвлённую систему воздушных труб и проходит в помещение через диффузоры, они нужны для того, чтобы можно было регулировать механически проход воздуха через конкретное проходное отверстие, ведь иначе через выходные отверстия, которые ближе к калориферу будет проходить больше воздуха.

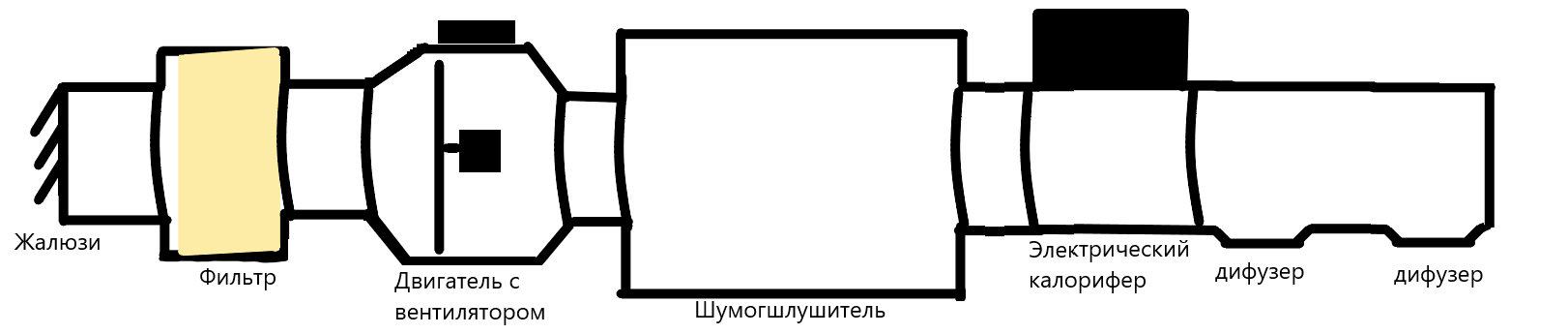


Рис. 12 Вид системы со всеми компонентами без контроллера

Вывод по главе 1

В качестве исполнительного устройства был выбран электрический калорифер. Датчик для влажности DHT22, для температуры DS18B20. Используя контроллер Arduino Mega систему можно масштабировать на большое количество помещений со своими исполнительными устройствами и датчиками температур и влажности, обработкой сможет заниматься один контроллер, вывод информации и управлении может быть реализован как на одном, так и на нескольких дисплеях. Также в различных помещениях возможно исполнение разных принципов управления. В связи с этим мы можем говорить о чрезвычайной гибкости системы при малой стоимости, что говорит о экономической целесообразности такой системы.

1. Разработка системы, алгоритмов управления и интерфейса
2. Алгоритм управления для поддержки температуры

Хоть электрический калорифер не так инерционен, как водяной калорифер объём воздуха, даже в малых помещениях весьма значителен, поэтому можно говорить, что система инерциальная. В таком случае можно применить простейшее on/off регулирование, как в двухпозиционных регуляторах. В системе будет 3 состояния:

* Бездействие
* Охлаждение
* Нагревание

Состояние бездействия – это моменты времени, когда система лишь замеряет температуру и отображает её. Система находится в этом состоянии если выполняются условие: температура в помещении (ТД) равна температуре уставки (Туст) с гистерезисом (Н) () и не включён двигатель вентилятора.

Состояние охлаждения – система находится в этом состоянии пока выполняются условие: температура в помещении (ТД) больше температуре уставки (Туст) плюс гистерезис (Н) () и запущен двигатель вентилятора.

Состояние нагревания – момент времени, когда включается электрический калорифер. Система находится в этом состоянии пока выполняются условие: температура в помещении (ТД) меньше температуре уставки (Туст) минус гистерезис (Н) (), а также запущен двигатель вентилятора.

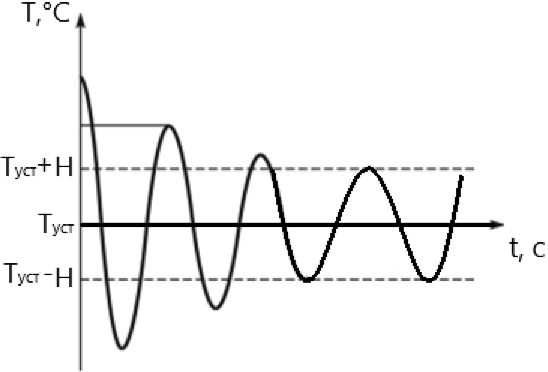


Рис. 13 график поведения системы

1. Интерфейс системы

Интерфейс системы, реализованный на дисплее: сенсорное управление экрана-резистивное, т.е. при нажатии мы можем фиксировать напряжение, выдаваемое по вертикали и горизонтали. Таким образом реализовано задание уставки, а именно: при нажатии или зажатии на кнопку DNt температура уставки понижается, на UPt повышается. Значение уставки сохраняется в ПЗУ Arduino, таким образом при перезагрузке устройства значение уставки сохраняется. Помимо этого, присутствуют клавиши UPv и DNv, которые отвечают за режим работы двигателя вентилятора, при увеличении этого значения на автотрансформатор подаётся сигнал на соответствующую обмотку и увеличивается напряжение, подаваемое на двигатель, что в свою очередь увеличивает количество оборотов вентилятора.

Также на экране отображается значение текущей температуры, влажности, уставки и режима работы двигателя. Ниже происходит отрисовка графика содержащий в себе линию уставки, легенду температуры и влажности (81 точка), обновление измерений происходит раз в 1,5 секунды, таким образом мы можем видеть динамику температуры за время около двух минут. На Рис. 14 представлена фотография система, некоторое время прорабатывавшая, для удобства есть сигнальный светодиоды, сигнализирующие о режиме работы двигателя и необходимости включения калорифера, также имеется мигающий светодиод для понимания, что система не зависла.

В интерфейсе системы реализована ситуация, когда температура опускается до 0, тогда происходит перемасштабирование графика – его диапазон изменяется от -50 до +50 градусов, при этом вся легенда сохраняется, спустя 2 минуты (когда закончились точки графика ниже 0), если больше не было температуры ниже 0 график обратно масштабируется от 0 до 50 градусов.

Текущее значение температуры отображается красным, как и график. Уставка – белая, влажность – синяя.

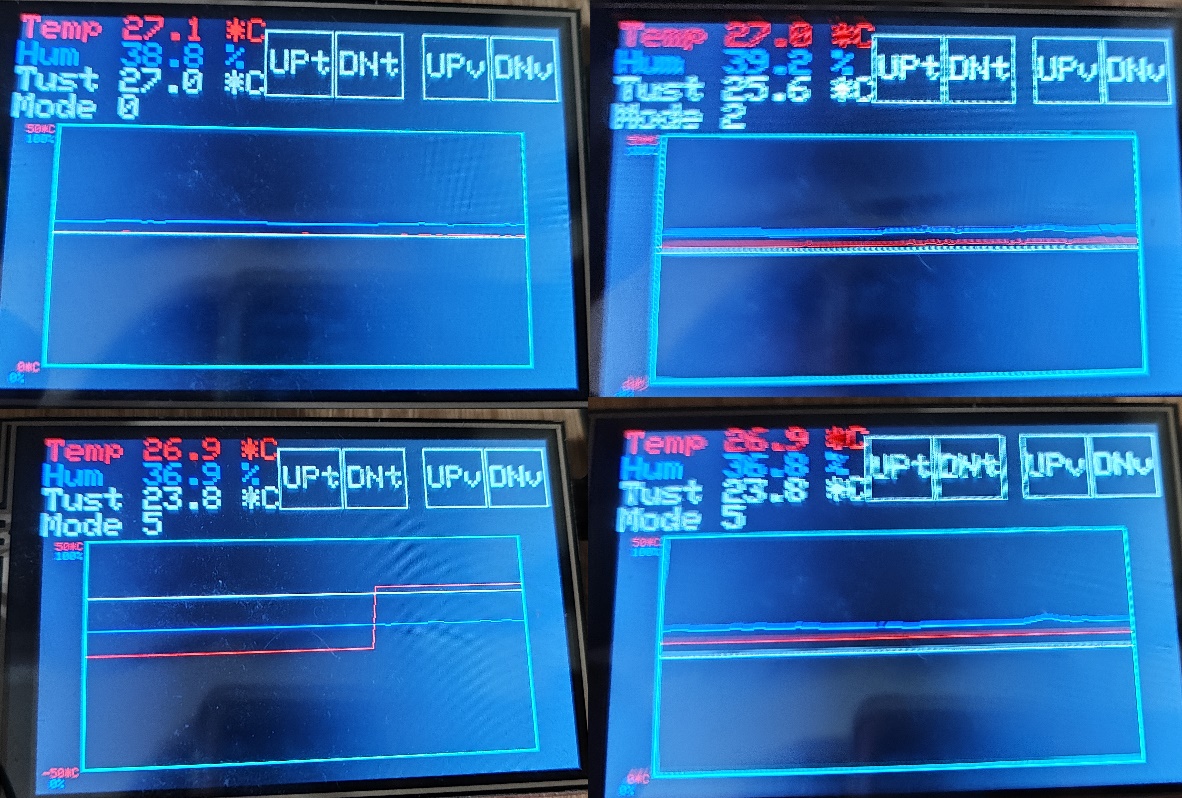


Рис. 14 фотографии работы системы

Часть текста является неизменной и отрисовывается в момент инициализации, но графики и значения вcё время меняются поэтому вызываются процедуры для их отрисовки. Настройка текста подразумевает собой установку курсора, выбор цвета и размера текста. TF1 флаг масштабирования, если присутствуют отрицательные температуры Nism константа количества измерений (точек). Для отрисовки чёрных прямоугольников необходимо задать координаты, эти области обновлений статичный поэтому задаются всегда одинаковые координаты. Для отрисовки линий необходимо задать координаты двух точек, которые постоянно меняются, для этого создан массив, где с каждым снятым значением самое старое удаляется, а новое записывается, причём все значения в массиве сдвигаются. Формула для отрисовки линий графика влажности всегда одинаковая и не зависит от масштабирования в отличии от температуры. Курсор всегда стоит в точке (0;0), что соответствует левому верхнему углу дисплея, а размерность области графиков 400 на 200 пикселей, так что в формулах задаются смещения и для графиков температуры некоторые значения приходится округлять до целых. Длинна линий между пикселями 5 точек. Номер объекта в массиве i, sH массив значений влажности, sT массив значений температуры. Циклом проходимся всем значениям в массивах и с каждым циклом отрисовываем линию. Расчёт точек для влажности:



Расчёт точек для температуры без масштабирования (от 0 до 50 градусов):



Расчёт точек для температуры с масштабированием (от -50 до +50 градусов). Здесь значений могут получатся не целые, а координаты каждого пикселя целая число, поэтому приводим к целочисленному значению.



Формулы для отрисовки уставки без масштабирования, Tust это задаваемая пользователем величина температуры уставки, хранящаяся в ПЗУ.



Формулы для отрисовки уставки с масштабированием:



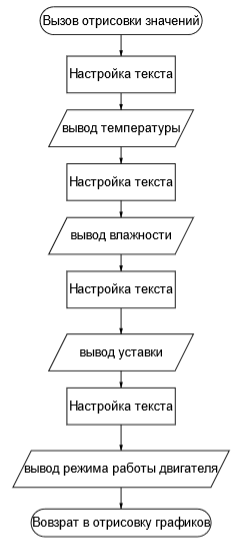


Рис. 15 блок схема отрисовки значений

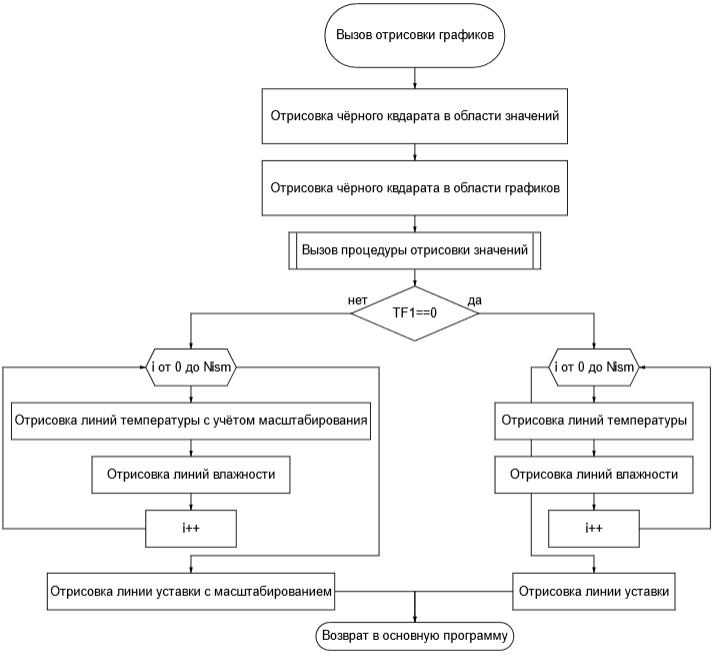


Рис. 16 блок схема процедуры отрисовки графиков

1. Алгоритм программной реализации микроконтроллера

Изначально в контролере подключаются необходимые библиотеки, после объявляются все необходимые константы, переменные и объекты. После начинается процесс инициализации, т.е. код, который запускается один раз при включении устройства, в нём происходит:

* Чтение из ПЗУ температуры уставки.
* Инициализация дисплея.
* Настройка пинов, необходимые пины: экрана, мигающего светодиода, светодиод охлаждения и обогрева, светодиоды режимов вентиляции, датчиков.
* Проводим настройки экрана: задаём положение, устанавливаем курсоры, цвет, размер текста, выводим неизменяемый текст на экран.
* Обрисовываем кнопки.
* Проводим первичные измерения значений температуры и влажности.
* Обрисовываем график и его границы.

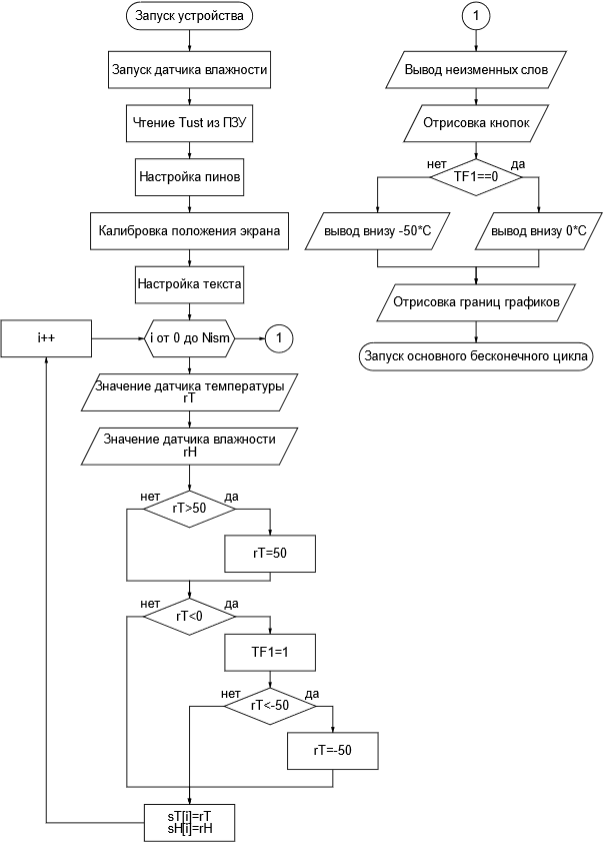


Рис. 17 Блок схема инициализации

Далее идёт основной бесконечный цикл:

* Задаём таймер в 1.5 секунды для мигающего светодиода.
* Проверяем напряжение на экране, для того чтобы понять нажата ли кнопка, если да, то меняем температуру уставки или режим работы двигателя. И записываем в ПЗУ если значение изменилось. Также происходит проверка, чтобы пользователь не выходил из диапазона допустимых значений температуры уставки и режимов работы.
* Если режим работы изменился, то подаём сигнал на соответствующий пин, остальные пины коммутации отключаем.
* Опрашиваем датчики.
* Если температура стала отрицательной, масштабируем график и включаем флаг, запускаем счётчик и обнуляем его.
* Если есть флаг масштабирования и не было отрицательного значения увеличиваем счётчик.
* Если счётчик достиг максимального значения снимаем флаг и убираем масштабирование.
* Отрисовываем графики значений температуры, влажности и уставу.
* Проверяем температуры относительно уставки и гистерезиса, в соответствии со значением включаем, если необходимо калорифер.

Millis встроенная функция, которая отчитывает миллисекунды с момента запуска устройства. LT переменная счётчик. Экспериментально было подобрано, что принимать изменения, при нажатии стоит в течении полутра секунд, а остальное время их игнорировать, так как иначе почти невозможно установить желаемые величины, также некоторое время уходит на отрисовку, и во время неё Arduino также не может обрабатывать нажатия. В данном дисплее не представляется возможным обрабатывать их по прерыванию, однако ели установить другие кнопки (например, механические), то проблему можно решить обработкой по прерываниям. Также из-за резистивного экрана пришлось вводить «мёртвые зоны», так как даже при отсутствии нажатия всегда выдавалось некоторое напряжение, которое могло изменять значения, как при нажатии. Для упрощения программы обработка нажатий реализована только по оси Y, игнорируя напряжение с оси Х. Для подобных дисплеев «мёртвые зоны» необходимо подбирать индивидуально для каждого устройства. Одним из вариантов решения проблем перевод всех кнопок на механическое управление, либо использования другого дисплея. Delay также является встроенной функцией реализующей задержку всего контроллера, принимает значение в миллисекундах, рекомендуется ставить при обработке различных нажатий. 31 пин-мигающий светодиод. 33 пин-включение вентилятор и запуск привода жалюзи.41 пин-режим 0, 43-режим 1, 45-2, 47-3, 51-4, 53-5. Реле переключается при значении сигнала LOW.

Весь код контроллера со всеми функциями представлен в [приложении А](#ПриложениеА).

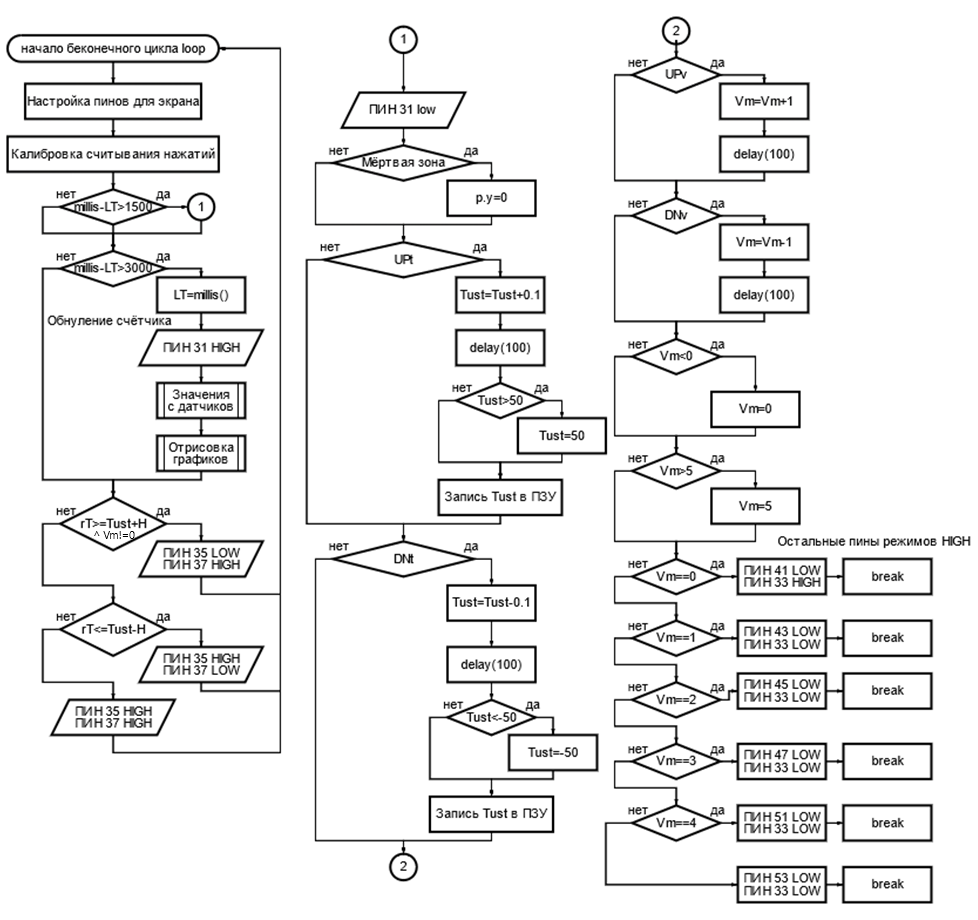


Рис. 18 Блок схема основного бесконечного цикла

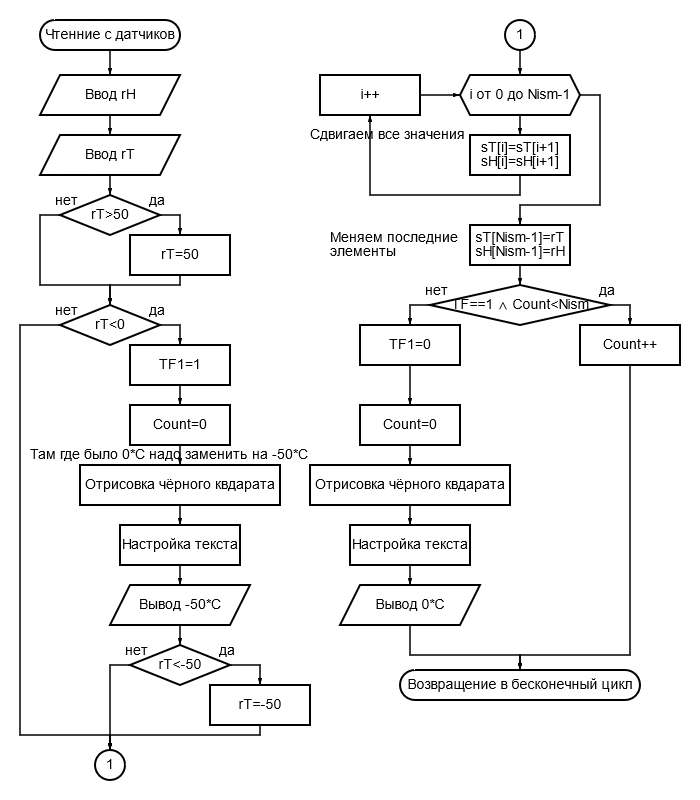


Рис. 19 Блок схема чтения значений

Вывод по главе 2

В системе используется двухпозиционное регулирование (включение или выключении калорифера). Предусмотрено масштабирование графиков в случае отрицательных температур и контролируется выход за пределы диапазонов. Для управления системой имеются четыре сенсорные кнопки, отвечающие за повышение, понижение температуры уставки, записи их в ПЗУ, для сохранения настроенных значений в случае перезапуска устройства, а также кнопки, меняющие режим работы двигателя вентилятора. Таким образом реализован пользовательский интерфейс.

1. Схема подключения оборудования
2. Блок реле

Для управления устройствами использовался специальный блок реле, состоящий из 16 реле низкого уровня «Модуль реле 16-канальный 10А250В 12V на базе LM2596 (Low Level Trigger)». Модуль реле с 16 каналами, оснащенными опторазвязкой для обеспечения изоляции между силовой логической частями. Управление модулем осуществляется с помощью TTL сигнала напряжением 5В. Каждый канал реле имеет отдельную линию управления и активируется при подаче низкого TTL уровня.

Данный модуль разработан для управления силовыми нагрузками, такими как моторы, освещение и другие устройства, с помощью микроконтроллеров Arduino, ESP или STM. На плате модуля присутствует светодиодная индикация, позволяющая визуально определить состояние каждого реле.

Каждое реле способно выдержать нагрузку до 10А при напряжении AC220В. Для питания модуля реле требуется источник питания напряжением 12В, снабженный не менее чем 1.5А. Кроме того, на плате предусмотрено место для установки стабилизатора на базе микросхемы LM2596.

В целом, данный модуль реле обладает высокой функциональностью и позволяет легко и удобно управлять различными силовыми нагрузками с помощью микроконтроллеров. Он может быть эффективно применен в различных проектах, требующих надежного и гибкого управления силовыми устройствами.

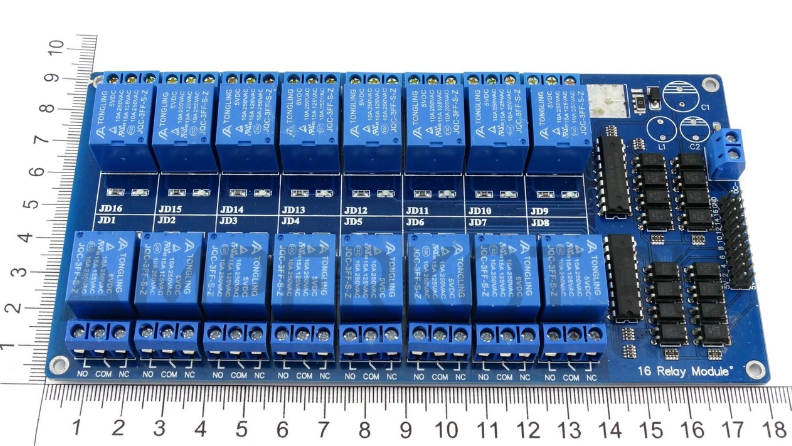


Рис. 20 Модуль реле

1. Подключение дисплея

Дисплей в свою очередь подключается непосредственно на плату, занимая почти все ПИНы, однако благодаря этому нет нужды для дополнительной макетной платы, чтобы подключить его к Arduino. Ниже представлена схема подключения дисплея к плате. Из неё можно увидеть, что она перекрывает некоторые ПИНы, которые не используются, что в свою очередь уменьшает количество доступных ПИНов, однако у Arduino Mega сбоку имеется большое количество не перекрытых ПИНов, в том числе питания и земли.

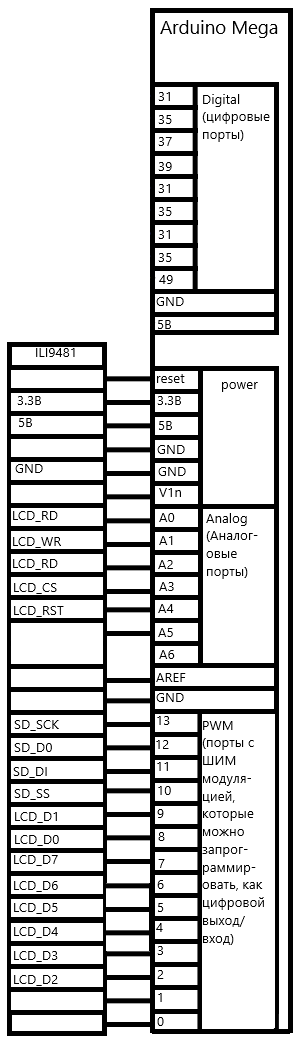


Рис. 21 Схема подключения дисплея к плате

1. Схема подключения датчиков

На свободные ПИНы, были подключены канальный высокоточный датчик температуры DS18B20 и датчик температуры и влажности DHT22, предоставленные на Рис. 22.

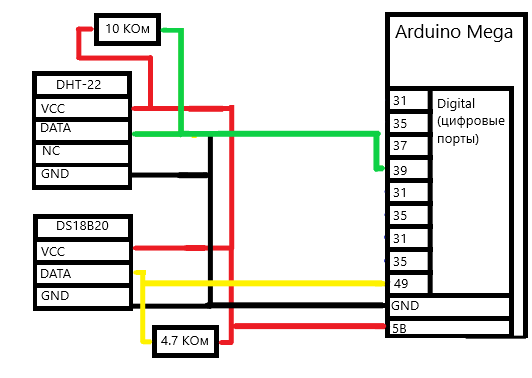


Рис. 22 схема подключения датчиков

1. Подключение исполнительных устройств

В связи с отсутствием объекта управления, для оценки работы устройства используется вышеописанный блок реле, к которому устройства и подключаются, так как каждое реле имеет свой светодиод, можно легко судить о включении и отключении конкретного устройства. Все реле имеют свои номера 1 реле отвечает, за коммутирование автотрансформатора на землю. 2,3,4,5 и 6 реле за различные режимы работы двигателя от малых оборотов до высоких оборотов (по сути, эти реле заменяют автотрансформатор). 16 реле отвечает за привод жалюзи, и оно коммутируется в случае, если активно со 2 по 6 реле. 15 реле коммутирует питание на электрический калорифер, разумеется, только в случае если активен вентилятор и открыты жалюзи. На Рис. 23 слева сверху 15 и 16 реле. Слева снизу 1-6. За активацию 16 реле отвечает низкий сигнал на 33 ПИНе, за 15 реле 37 ПИН, 1-6 реле 41,43,45,47,51,53 соответственно.



Рис. 23 Работа реле

Вывод по главе 3

Рассмотрена схема подключение всех компонентов, а также сигналов приводящих к запуску исполнительных устройств, и сигналов сигнализирующий о работе системы. Описан блок реле и его работа.

1. Первоначальное тестирование
2. Общая схема тестирования

Изначально тестирование всех сигналов производилось по средствам измерения мультиметром и постановке блока светодиодов на макетную плату. Подавались различные сигналы на ПИНы, переменные температуры и влажности задавались вручную, путём генерации случайных значений, в зависимости от температуры должны загорятся либо затухать светодиоды. В последствии блок светодиодов было решено оставить в качестве индикации, ведь, по сути, эти же сигналы ведут на блок реле, который уже в свою очередь коммутирует различные устройства. Тестирование блока реле представлялось наиболее простым, в связи с характерным щелчком переключения механического реле и наличия у них таких же светодиодов, как и у изначального блока оных.

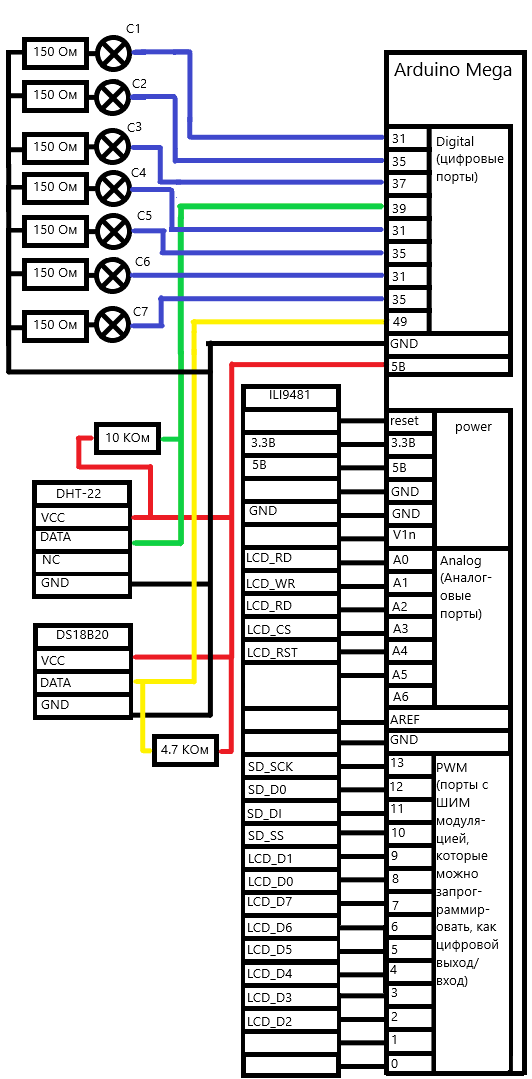


Рис. 24 Первоначальная схема тестирования сигналов

1. Тестирование датчиков

С появлением датчиков отпала необходимость генерировать случайные числа и значения брались непосредственно их опросом, и сверялись с обычным комнатным термометром для проверки достоверности кода опроса. Также была необходимость в масштабировании графиков в случае отрицательных температур помещения, для этого были использована генерация случайных отрицательных значений массива температур, после настройки необходимо было проверить работоспособность на датчиках, для этого использовались стаканы с холодной водой, а также льдом. Датчик DS18B20 имеет герметичный корпус, благодаря чему было, возможно не опасаясь выхода из строя оного, проводить тестирования холодной водной среде.

1. Тестирование экрана

Тестирование и настройка экрана была наиболее трудоёмким процессом, в связи с отсутствием нормальной документации. Приходилось настраивать его по примерам, идущих в комплекте с датчиком. В целом отрисовка графики не вызвала сложностей, в отличии от обработки нажатий, в силу устройства резистивного экрана. Встроенные функции выдавали некоторое закодированное значения напряжения и для определения «мёртвых зон» и границ кнопок приходилось на этом же дисплее выводить код выдаваемого напряжения по конкретной оси и записывать зависимость передвижения стилуса по экрану и напряжения по оси, так был подобран диапазон для каждой из кнопок.

Вывод по 4 главе

Рассмотрены способы тестирования сигналов и значений компонентов. Была приведена схема, подключения светодиодов для проверки ПИНов. Описан методы настройки нажатий на дисплей, и проверки значений датчиков.

ЗАКЛЮЧЕНИЕ

Целью данной выпускной квалификационной работы была разработка системы вентиляции для офисных помещений. На этапе выполнения ВКР удалось выбрать необходимые для работы системы компоненты. Были выбраны и реализованы алгоритмы управления системой, а также сделан пользовательский интерфейс на сенсорной панели для работы с системой. Рассмотрены различные виды канальных нагревателей и для работы системы был выбран электрический калорифер.

Процесс разработки данной системы показал, что необходимо учитывать большое количество нюансов, при реализации системы для каждого отдельного объекта, чтобы конечная система удовлетворяла полностью требования заказчика.

СПИСОК ЛИТЕРАТУРЫ

1. Документация на микроконтроллер Arduino Mega 2560 [Электронный ресурс]. – Режим доступа: <https://docs.arduino.cc/static/d52e67df6975cefd79511c206267486e/A000067-datasheet.pdf>
2. Виктор Петин: Издательство BHV; Проекты с использованием контроллера Arduino. [книга], 2021 г.
3. Нимич Г.В., Михайлов В.А., Бондарь Е.С. Издательство: ТОВ Видавничий будинок Аванпост-Прим; Современные системы вентиляции и кондиционирования воздуха [книга], 2003 г.
4. НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ ЧИСТЫЕ ПОМЕЩЕНИЯ Вентиляция и кондиционирование воздуха Общие требования. [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200124954>

ПРИЛОЖЕНИЕ А

//Библиотеки

#include <Adafruit\_GFX.h> // Core graphics library

#include <Adafruit\_TFTLCD.h> // Hardware-specific library

#include <TouchScreen.h>

#include "DHT.h"//датчик влажности

#include <OneWire.h>//датчик температуры

#include <EEPROM.h> //библиотека для ПЗУ

#if defined(\_\_SAM3X8E\_\_)

#undef \_\_FlashStringHelper::F(string\_literal)

#define F(string\_literal) string\_literal

#endif

//константы для экрана

// For better pressure precision, we need to know the resistance

// between X+ and X- Use any multimeter to read it

// For the one we're using, its 300 ohms across the X plate

#define YP A3 // must be an analog pin, use "An" notation!

#define XM A2 // must be an analog pin, use "An" notation!

#define YM 9 // can be a digital pin

#define XP 8 // can be a digital pin

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

#define LCD\_CS A3 // Chip Select goes to Analog 3

#define LCD\_CD A2 // Command/Data goes to Analog 2

#define LCD\_WR A1 // LCD Write goes to Analog 1

#define LCD\_RD A0 // LCD Read goes to Analog 0

#define LCD\_RESET A4 // Can alternately just connect to Arduino's reset pin

#define BLACK 0x0000

#define BLUE 0x001F

#define RED 0xF800

#define GREEN 0x07E0

#define CYAN 0x07FF

#define MAGENTA 0xF81F

#define YELHIGH 0xFFE0

#define WHITE 0xFFFF

#define DHTPIN 39

#define DHTTYPE DHT22

Adafruit\_TFTLCD tft(LCD\_CS, LCD\_CD, LCD\_WR, LCD\_RD, LCD\_RESET);

//константы

const word Nism=81;//количество точек измерений

const word HustT=1;//гистерезис уставки

const byte TustAdr=3;

//переменные

word Count=0;//счётчик для сброса автомасштабирования

word Vm=0;//Vmode режим вентилятора по умолчанию 0, вентилятор выключен

float Tust=23;//температура уставки

float rT=Tust;//температура с датчика

float rH=Tust;//влажность с датчика

float sT[Nism]={};//сохранённые значения температуры

float sH[Nism]={};//сохранённые значения влажности

unsigned long last\_time=0;//последнее время

byte TFl=0;//температурный флажок на изменение масшатаба

TSPoint p;

DHT dht(DHTPIN,DHTTYPE);//Создаем объект dht

OneWire ds(49); // Создаем объект OneWire для шины 1-Wire, с помощью которого будет осуществляться работа с датчиком

//начало настройки/////////////////////////////////////////////////////

void setup(void) {

dht.begin();//запуск датчика влажности

Tust=(float)EEPROM.read(TustAdr);//Чтение температуры уставки из ПЗУ

//инициализация дисплея

tft.reset();

uint16\_t identifier = 0x9341;

tft.begin(identifier);

//настройка пинов

pinMode(13, OUTPUT);//для экрана

pinMode(31,OUTPUT);//Мигающий светодиод (3с)

//pinMode(33,OUTPUT);//Всегда горящий светодиод

pinMode(35,OUTPUT);//Светодиод когда работает охлаждение

pinMode(37,OUTPUT);//Светодиод когда работает обогрев

pinMode(41,OUTPUT);//Светодиод режима вентиляции 0

pinMode(43,OUTPUT);//Светодиод режима вентиляции 1

pinMode(45,OUTPUT);//Светодиод режима вентиляции 2

pinMode(47,OUTPUT);//Светодиод режима вентиляции 3

pinMode(51,OUTPUT);//Светодиод режима вентиляции 4

pinMode(53,OUTPUT);//Светодиод режима вентиляции 5

pinMode(33,OUTPUT);//Привод жалюзи

//настройка экрана

tft.setRotation(1);//положение экрана

tft.fillScreen(BLACK);//заполняем всё чёрным

tft.setCursor(0, 0);//установка курсора в начало

tft.setTextColor(WHITE); tft.setTextSize(5);//выбираем цвет и размер текста

tft.println("Get first mesuaring");//выводим на экрна слова

tft.println("Please wait");

//Проводим первые измерения, чтоб изначально построить графики

for(int i=0;i<Nism;i++){

//rT=random(-2,-1);

rT=dht.readTemperature();

rH=dht.readHumidity();

if(rT>50){//Если значене больше 50 приравниваем к 50

rT=50;

}

if(rT<0){

TFl=1;

if(rT<-50){//Если значене меньше -50 приравниваем к -50

rT=-50;

}

}

//rH=random(20,60);

sT[i]=rT;

sH[i]=rH;

}

//Выводим на экран неизменные слова

tft.fillScreen(BLACK);

tft.setCursor(0, 0);

tft.setTextColor(RED); tft.setTextSize(3);

tft.println("Temp ");

tft.setTextColor(BLUE); tft.setTextSize(3);

tft.println("Hum ");

tft.setTextColor(WHITE); tft.setTextSize(3);

tft.println("Tust ");

tft.setTextColor(WHITE); tft.setTextSize(3);

tft.println("Mode ");

tft.setTextColor(RED); tft.setTextSize(1);

tft.setCursor(0, 100);

tft.println(" 50\*C");

tft.setTextColor(BLUE); tft.setTextSize(1);

tft.println(" 100%");

//Отрисовка конопок

tft.setCursor(219, 28);

tft.setTextColor(WHITE); tft.setTextSize(3);

tft.println("UPt");

tft.drawRect(214, 10, 60, 60, WHITE);

tft.setCursor(280, 28);

tft.println("DNt");

tft.drawRect(276, 10, 60, 60, WHITE);

tft.setCursor(358, 28);

tft.println("UPv");

tft.drawRect(354, 10, 60, 60, WHITE);

tft.setCursor(420, 28);

tft.println("DNv");

tft.drawRect(416, 10, 60, 60, WHITE);

tft.setCursor(0, 300);

tft.setTextColor(RED); tft.setTextSize(1);

if(TFl==0){

tft.println(" 0\*C");

}

else{

tft.println(" -50\*C");

}

tft.setTextColor(BLUE); tft.setTextSize(1);

tft.println(" 0%");

tft.drawRect(39, 99, 403, 203, CYAN);//отрисовка границ графика

}

//Начало основного цикла/////////////////////////////////////////////////////

void loop() {

digitalWrite(13, LOW);//необходимо для работы экрана

TSPoint p = ts.getPoint();//получаем значение нажатых пикселей

digitalWrite(13, HIGH);//необходимо для работы экрана

pinMode(XM, OUTPUT);//необходимо для работы экрана

pinMode(YP, OUTPUT);//необходимо для работы экрана

p.y = map(p.y, 0, 920, 480, 0);

//digitalWrite(33,LOW);//Всегда горящий светодиод

if (millis()-last\_time>1500){

digitalWrite(31,LOW);//Мигающий светодиод (3с)

if( p.y>110 && p.y<130){//мёртвая зона, так как по умолчанию эти значения

p.y=0;

}

if (p.y>170 && p.y<250){//UPt

Tust=Tust+0.1;

delay(100);

if(Tust>50)//ели уставка больше 50 градусов, то приравниваем к 50

{Tust=50;}

EEPROM.write(TustAdr, (float)Tust);//Запись Туст в пзу

}

if (p.y>130 && p.y<165 ){//DNt

Tust=Tust-0.1;

delay(100);

if(Tust<-50)//ели уставка меньше -50 градусов, то приравниваем к -50

{Tust=-50;}

EEPROM.write(TustAdr, (float)Tust);//Запись Туст в пзу

}

if (p.y>42 && p.y<79){//UPv

Vm=Vm+1;

delay(100);

}

if (p.y>0 && p.y<38){//DNv

Vm=Vm-1;

delay(100);

}

}

if(Vm<=0)//ели режим меньше 0 то 0

{Vm=0;}

if(Vm>=5)//ели режим больше 5 то 5

{Vm=5;}

switch(Vm){//переключатель режима дивгателя

case 0:

digitalWrite(41,LOW);

digitalWrite(43,HIGH);

digitalWrite(45,HIGH);

digitalWrite(47,HIGH);

digitalWrite(51,HIGH);

digitalWrite(53,HIGH);

digitalWrite(33,HIGH);

break;

case 1:

digitalWrite(41,HIGH);

digitalWrite(43,LOW);

digitalWrite(45,HIGH);

digitalWrite(47,HIGH);

digitalWrite(51,HIGH);

digitalWrite(53,HIGH);

digitalWrite(33,LOW);

break;

case 2:

digitalWrite(41,HIGH);

digitalWrite(43,HIGH);

digitalWrite(45,LOW);

digitalWrite(47,HIGH);

digitalWrite(51,HIGH);

digitalWrite(53,HIGH);

digitalWrite(33,LOW);

break;

case 3:

digitalWrite(41,HIGH);

digitalWrite(43,HIGH);

digitalWrite(45,HIGH);

digitalWrite(47,LOW);

digitalWrite(51,HIGH);

digitalWrite(53,HIGH);

digitalWrite(33,LOW);

break;

case 4:

digitalWrite(41,HIGH);

digitalWrite(43,HIGH);

digitalWrite(45,HIGH);

digitalWrite(47,HIGH);

digitalWrite(51,LOW);

digitalWrite(53,HIGH);

digitalWrite(33,LOW);

break;

case 5:

digitalWrite(41,HIGH);

digitalWrite(43,HIGH);

digitalWrite(45,HIGH);

digitalWrite(47,HIGH);

digitalWrite(51,HIGH);

digitalWrite(53,LOW);

digitalWrite(33,LOW);

break;

}

if (millis()-last\_time>3000){//Снимаем данные и отрисовываем раз в полторы секунды

last\_time=millis();//обнуляеем счётчик

digitalWrite(31,HIGH);//Мигающий светодиод (3с)

ReadData(); //процедура для чтения информации с датчика DHT22 и DS18B20

DrawGrapth();//Процедура для отрисовки графиков и измерений

}

if(rT>=Tust+HustT ){//если считанная температура превышает установленную + гистерезис

digitalWrite(35,LOW);

digitalWrite(37,HIGH);

}

else if((rT<=Tust-HustT) && (Vm!=0) ){//если считанная температура ниже установленной - гистерезис

//включаем обогрев

digitalWrite(37,LOW);

digitalWrite(35,HIGH);

}

else{//если всё хорошо выключаем вентилятор,закрываем жалюзи и обогрев

digitalWrite(37,HIGH);

digitalWrite(35,HIGH);

}

}

//Процедуры://///////////////////////////////////////////////////

void ReadData(){//Процедура считывающая значения с датчика

//читаем с датчика

rH = dht.readHumidity();//Читаем влажность

byte data[2]; // Место для значения температуры

ds.reset(); // Начинаем взаимодействие со сброса всех предыдущих команд и параметров

ds.write(0xCC); // Даем датчику DS18b20 команду пропустить поиск по адресу. В нашем случае только одно устрйоство

ds.write(0x44); // Даем датчику DS18b20 команду измерить температуру. Само значение температуры мы еще не получаем - датчик его положит во внутреннюю память

delay(10); // Микросхема измеряет температуру, а мы ждем.

ds.reset(); // Теперь готовимся получить значение измеренной температуры

ds.write(0xCC);

ds.write(0xBE); // Просим передать нам значение регистров со значением температуры

// Получаем и считываем ответ

data[0] = ds.read(); // Читаем младший байт значения температуры

data[1] = ds.read(); // А теперь старший

// Формируем итоговое значение:

// - сперва "склеиваем" значение,

// - затем умножаем его на коэффициент, соответсвующий разрешающей способности (для 12 бит по умолчанию - это 0,0625)

rT = ((data[1] << 8) | data[0]) \* 0.0625;

if(rT>50){//если больше 50 то 50

rT=50;

}

if(rT<0){//если меньше 0 поднимаем флаг маштабирования и меняем значение внизу

TFl=1;

Count=0;

tft.fillRect(0, 280, 30, 300, BLACK);//заполняем нижнюю границу, где было 0\*С

tft.setTextColor(RED); tft.setTextSize(1);

tft.setCursor(0, 300);

tft.println(" -50\*C");

if(rT<-50){//Если значене меньше -50 приравниваем к -50

rT=-50;

}

}

//Сдвигаем все значения из легенды влажности и температуры на 1 влево

for(int i=0;i<(Nism-1);i++){

sT[i]=sT[i+1];

sH[i]=sH[i+1];

}

//Меняем последние элементы

sT[Nism-1]=rT;//В поледний элемент массива записываем новую температуру

sH[Nism-1]=rH;//В поледний элемент массива записываем новую влажность

if(TFl==1 && Count<Nism){//если масштабируем и массив не перезаписался, то увеличиваем счётчие

Count++;

}

else{//иначе убираем автомасштабирование

TFl=0;

Count=0;

tft.setTextColor(RED); tft.setTextSize(1);

tft.fillRect(0, 280, 30, 300, BLACK);//заполняем нижнюю границу, где было -50\*С

tft.setCursor(0, 300);

tft.println(" 0\*C");

tft.setTextColor(BLUE); tft.setTextSize(1);

tft.println(" 0%");

}

}

void DrawGrapth(){//Процедура отрисовки графиков и значений

tft.fillRect(70, 0, 144, 93, BLACK);//Запалняем область значений чёрным квадратом

tft.fillRect(40, 100, 401, 201, BLACK);//Запалняем область графиков чёрным квадратом

DrawIzm();//Процедура отрисовки значений

if(TFl==0){

for(int i=0; i<(Nism-1); i++){//строим графики по Nism точкам. имеем по x 400px по y 200px.Относительно курсора (0;0)

tft.drawLine(40+5\*i, -4\*sT[i]+300, 5\*i+45, -4\*sT[i+1]+300, RED);//график температуры

tft.drawLine(40+5\*i, -sH[i]/2+200, 5\*i+45, -sH[i+1]/2+200, BLUE);//график влажности

}

tft.drawLine(40, -4\*Tust+300, 440, -4\*Tust+300, WHITE);//рисуем линию уставки

}

else{

for(int i=0; i<(Nism-1); i++){//строим графики по Nism точкам. имеем по x 400px по y 200px.Относительно курсора (0;0)

tft.drawLine(40+5\*i, int(-2\*sT[i]+200), 5\*i+45, int(-2\*sT[i+1]+200), RED);//график температуры

tft.drawLine(40+5\*i, -sH[i]/2+200, 5\*i+45, -sH[i+1]/2+200, BLUE);//график влажности

}

tft.drawLine(40, int(-2\*Tust+200), 440, int(-2\*Tust+200), WHITE);//рисуем линию уставки

}

}

void DrawIzm(){//Процедура отрисовки значений

tft.setCursor(0, 0);//Установка курора в левый верхний угол

tft.setTextColor(RED); tft.setTextSize(3);//Выбор цвета и размера текста

tft.print(" ");//пробелы так, как слева имеем постоянные слова Temp,Hum,Tust

tft.print(rT,1);//Выводим последнюю полученную температуру

tft.println(" \*C");//Выводим цельсий

//tft.println(p.y);

tft.setTextColor(BLUE); tft.setTextSize(3);

tft.print(" ");

tft.print(rH,1);//Выводим последнюю полученную влажность

tft.println(" %");

tft.setTextColor(WHITE); tft.setTextSize(3);

tft.print(" ");

tft.print(Tust,1);//Выводим температуру уставки

tft.println(" \*C");

tft.print(" ");

tft.print(Vm,1);//Выводим режим работы двигателя

}