



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименование института полностью)

Кафедра Промышленная электроника  
(наименование)

**ЗАДАНИЕ**  
**на выполнение бакалаврской работы**

Студент Минеев Алексей Сергеевич, Элб-1601а

1. Тема ПИД-регулятор температуры

2. Срок сдачи студентом законченной выпускной квалификационной работы \_\_\_\_\_

3. Исходные данные к выпускной квалификационной работе:

Проверка температуры окружающего воздуха

Регулирование температуры окружающего воздуха

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов)

Аннотация

Введение

1. Состояние вопроса

2. Аппаратная часть

3. Программная часть

Заключение

Список используемой литературы

5. Ориентировочный перечень графического и иллюстративного материала

1. Обзор существующих решений

2. Схема структурная устройства

3. Схема электрическая принципиальная устройства

4. Блок-схема программы

5. Временные диаграммы ШИМ

6. Временные диаграммы ФР

6. Консультант по разделу Брега О.Н.

7. Дата выдачи задания «26» января 2020 г.

Руководитель выпускной  
квалификационной работы

\_\_\_\_\_  
(подпись)

Е.С. Глибин  
(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименование института полностью)

Кафедра Промышленная электроника  
(наименование)  
(наименование)

**КАЛЕНДАРНЫЙ ПЛАН**  
**выполнения бакалаврской работы**

Студент А.С. Минеев  
по теме ПИД-регулятор температуры

Наименование раздела (главы) <sup>1</sup>	Плановый срок выполнения раздела (главы) <sup>1</sup>	Фактический срок выполнения раздела (главы) <sup>1</sup>	Отметка о выполнении раздела (главы) <sup>1</sup>
Состояние вопроса	10.03.2020	10.03.2020	Выполнено
Обзор рынка	17.03.2020	17.03.2020	Выполнено
Аппаратная часть	05.04.2020	05.04.2020	Выполнено
Программная часть	28.04.2020	28.04.2020	Выполнено
Выбор компонентов устройства	12.05.2020	12.05.2020	Выполнено
Заключение	17.05.2020	17.05.2020	Выполнено
Список литературы	24.05.2020	24.05.2020	Выполнено

Руководитель бакалаврской  
работы

\_\_\_\_\_ (подпись)

Е.С. Глибин  
(И.О. Фамилия)

## Аннотация

Название выпускной квалификационной работы «ПИД-регулятор температуры».

Данная ВКР состоит из пояснительной записки на 44 страницы, введения на 2 страницы, включая 23 рисунка, 4 таблицы, списка 24 источников, в том числе 10 источников на иностранном языке и одного приложения.

ВКР может быть разделена на следующие логически взаимосвязанные части: состояние вопроса, проектная и программная части.

Мы начинаем с постановки задачи, а затем логически переходим к ее возможным решениям.

Сначала мы обсуждаем актуальность работы. Затем мы проводим анализ рынка. Далее мы рассматриваем исходные данные.

Мы сравнили существующие аналоги нашего проекта и поставили себе определенные цели и задачи.

Мы в общих чертах описываем соединение и программирование нашего устройства.

Подводя итоги, мы бы хотели подчеркнуть, что наш вариант ПИД-регулятора температуры оказался почти вдвое дешевле по сравнению с существующими аналогами. Для этого мы сравнили численные результаты стоимости продаваемых устройств и нашей суммарной сборки компонентов.

Таким образом, эти результаты приводят нас к выводу, что не всегда продаваемая вещь является самой дешевой.

## **Annotation**

The title of the final qualification work is “PID Temperature Controller”.

This WRC consists of an explanatory note of 44 pages, an introduction of 2 pages, including 23 figures, 4 tables, a list of 24 references, including 10 sources in a foreign language and one appendix.

WRC may be divided into several logically connected parts which are: status of the issue, design and program parts.

We start with the statement of the problem and then logically pass over to its possible solutions.

We first discuss the relevance of the work. Then we conduct a market analysis. Next, we consider the source data.

We compared the existing analogues of our project and set ourselves specific goals and objectives.

We outline the connection and programming of our device.

In conclusion we'd like to stress that our version of the PID temperature controller turned out to be almost half as much as compared to the existing analogs. To do this, we compared the numerical results of the cost of the devices sold and our total assembly of components.

Overall, the results suggest that the thing that is not always sold is the cheapest.

## Содержание

<b>Введение</b> .....	5
<b>1. Состояние вопроса</b> .....	7
1.1 Формулирование актуальности, цели и задач проекта.....	7
1.2 Рассмотрение начальных данных и обзор рынка.....	9
<b>2. Аппаратная часть</b> .....	15
2.1 Создание структурной схемы .....	15
2.2 Проектирование электрической принципиальной схемы .....	17
2.3 Подбор нужных элементов .....	19
<b>3. Программная часть</b> .....	39
3.1 Разработка алгоритма работы.....	39
3.2 Разработка кода программы .....	41
<b>Заключение</b> .....	44
<b>Список используемой литературы</b> .....	46
<b>Приложение А48 Код программы</b> .....	48

## Введение

Люди издавна стремились жить хорошо – убрать трудности, упростить каким-либо образом себе работу, сделать ее быстрее и качественнее, чтобы было больше времени на то, что они хотят. Одним из таких примеров был ПИД-регулятор.

Он был изобретен еще в 1910 году, однако лишь спустя 32 года, в 1942 году Джон Зиггер и Натаниэль Николс разработали метод его настройки. Однако долгое время они оставались достаточно дорогим и сложным устройством. Но с появлением в 1980-х годах первых микроконтроллеров их развитие и распространение стало происходить нарастающими темпами. И буквально каждые несколько лет число публикаций по данной теме росло: за первые 9 лет их число составило 14 штук, за следующие 9 – 111 штук, а за следующие 4 года их число выросло почти вдвое – 225 штук, в 2000 году на одном семинаре их представили сразу 90 штук, а количество патентов, зарегистрированных в патентной базе данных <http://gb.espacenet.com>, в начале 2006 года было уже 364.

По мнению многих экспертов наиболее распространенный тип регулятор именно ПИД-регулятор. Так порядка 90-95% регуляторов, которые используются в наши дни имеют ПИД-алгоритм.

Нам же наиболее интересен ПИД-регулятор температуры. Появление первого прибора для измерения температуры связывают с Г. Галилеем, который предложил такой прибор в 1598 г. Позднее М. Ломоносовым и Фаренгейтом были разработаны термометры, появились шкалы Цельсия, Кельвина, Ренкина, Фаренгейта. В дальнейшем люди делали так, чтобы температура измерялась точнее, и в соответствии с ней, они могли регулировать какие-либо процессы: начиная от банального открыть окно зимой, если в помещении стало жарко. До сложнейших процессов по созданию чего-либо в производстве, а также в лабораториях ученых.

Что же такое ПИД-регулятор температуры? Для начала мы каким-либо образом измеряем температуру и передаем ее значение ПИД-регулятору. Он высчитывает разницу между полученной температурой и заданной, умножает на коэффициенты, которые можно настроить, и выдает мощность, необходимую в данный момент. Однако это лишь одна из частей – пропорциональная (П). Она хороша своим быстродействием, однако не совсем точна. И как раз для устранения статической ошибки добавляется интегральная (И) составляющая. Она постепенно накапливает ошибку и лишь затем устраняет ее. И вот тут мы переходим к третьей части ПИД-регулятора – дифференциальной (Д). Она предназначена для улучшения быстродействия системы. В какой-то момент тормозя, а в другой ускоряя процесс, для лучшего регулирования. Существуют частные случаи ПИД-регулятора, состоящие из П, ПИ и ПД частей.



## **1. Состояние вопроса**

### **1.1 Формулирование актуальности, цели и задач проекта**

Поддержание температуры на необходимом уровне, а также ее своевременное регулирование является весьма важной частью большинства видов производств, не только крупных компаний, а также частных видов деятельности. В связи с этим ПИД-регулятор температуры, который помогает в этом вопросе, будет востребован и сможет решать поставленные перед ним задачи.

Он может быть востребован как в пищевой сфере: печи, термопрессы, оборудование для термоформинга и производства строительных материалов, системах отопления и водоснабжения, сушильных шкафах, холодильном оборудовании; так и пищевой: всевозможные инкубаторы, как для обычных яиц, всем знакомой курицы, так и для более экзотичных в наших краях яиц эублефаров – ящериц, чья популярность быстро растет, а также производства, использующие дрожжи. В частности, производство хлеба и пива.

Представьте, что необходимая температура для успешного завершения хода вашего начинания вдруг изменилась по разным причинам. Из-за этого яйца могут замерзнуть или свариться, а если говорить о дрожжах – то существует много разных видов, чья полезная нам деятельность находится в разных пределах температур. Например, если вид №1 более благоприятствует прохладному климату, а вид №2 – теплему, то из-за ненадлежащего контроля за температурой процесса может такое случиться, что они перестанут вести себя так как необходимо и на выходе мы получим не пойми что.

Для того, чтобы это не произошло подойдет ПИД-регулятор температуры. С помощью данного устройства, собранного на основе из схемы компании Arduino, на базе микроконтроллера Atmega328, возможно не

только регистрировать температуру в необходимом месте, но и регулировать ее своевременно и на высоком уровне точности.

Цель работы: создание ПИД-регулятора температуры, включающего в себя датчик температуры, а также элементы для детектирования уровня нуля сетевого напряжения и для управления самим нагревателем.

Задачи работы:

1. Изучить источники по рассматриваемой теме.
2. Проанализировать существующие варианты.
3. Создать структурную схему.
4. Спроектировать электрическую принципиальную схему.
5. Подобрать нужные элементы.
6. Написать код программы для устройства.
7. Создать и отладить модель ПИД-регулятора.

## 1.2 Рассмотрение начальных данных и обзор рынка

### Начальные данные

В своем проекте мы будем исходить из того, что плата Arduino имеет рекомендуемое входное напряжение 7-12 Вольт, а сама нагрузка в качестве нагревателя располагается будучи подключенной к сети переменного напряжения 220 Вольт 50 Герц.

ПИД-регулятор температуры будет выполнять следующие функции:

1. Опрос датчика температуры окружающего воздуха.
2. Просмотр на дисплее текущего значения.
3. Сравнение с заданным условием.
4. Регулирующее воздействие в зависимости от вышеназванного.

### Обзор рынка

Прежде чем начинать что-то придумывать стоит поискать – а вдруг такое уже было создано? И только не найдя искомое или найдя что-то похожее – придумать как можно улучшить, в частности используя более современные детали, оборудование, а также, при возможности, сделав новое устройство более дешевым, или с лучшими габаритами: вес, размер.

Так, если мы введем в сети интернет «ПИД-регулятор температуры», мы увидим множество ссылок, ведущих на разные сайты, предлагающие нам купить их продукцию по разной цене, всячески говоря, что ничего лучше, а также дешевле мы не найдем. Однако, если присмотреться, и используя фильтры, чтобы найти именно то, что нам необходимо, мы существенно уменьшим количество предлагаемых нам устройств. И уже их мы тщательно рассмотрим.

Цифровой контроллер температуры Maxwell PMA-49-R-2-96-N-N изображен на рисунке 1.

- Управление: реле
- Назначение: для управления электрическими нагревателями
- Выбор выходного сигнала: 9 типов термопары, термосопротивление, аналоговый выход
- ПИД регулирование, ограничение перерегулирования при подаче питания
- Четырехразрядный, двустрочный светодиодный дисплей
- Высокая точность измерений: 0,2% предела измерений
- Выходные сигналы: релейный, выход для управления твердотельным реле, аналоговый 4-20 мА, 0-5 В постоянного тока, 0-10 В постоянного тока, 1-5 В постоянного тока, тиристорный выход
- Отображения температуры по шкале Цельсия или Фаренгейта
- Питание: 90-260 В, 24 В (по запросу)
- Потребляемая мощность: 5 Вт
- Дисплей: Светодиодный четырехразрядный, двустрочный
- Точность регулирования:  $\pm 1$  °С
- Рабочая температура: 0 °С – 50 °С



Рисунок 1 - Maxwell PMA-49-R-2-96-N-N

ТРМ500 терморегулятор с мощным реле и крупным индикатором изображен на рисунке 2.

Терморегулятор ТРМ500 предназначен для поддержания температуры в таких областях промышленности как: печи, литье деталей из термопластов под давлением, переработке полимерного сырья до однородности, термическом переносе изображений на материалы, машинах для выдува ПЭТ-тары, в процессах по уменьшению неоднородности распределения веществ в объеме, запайщиках, оборудовании для изменения формы материала под давлением и температурой, в аппаратах по удалению влаги и другом оборудовании, для успешного получения результата на котором необходимо управление температурой.



Рисунок 2 - ОВЕН ТРМ500

Краткий список характеристик:

- ПИД-регулирование (в режиме нагревателя) или on/off-регулирование.
- Поддержка всех распространенных в России датчиков температуры.
- Внесен в реестр СИ с классом точности: ТС – 0,25 %, ТП – 0,5 %.
- Три выхода для управления нагрузкой или сигнализацией:
- Один или два индикатора (в зависимости от модификации).

- Увеличенный индикатор: высота цифр 20 мм.
- Питание: 96...264 В переменного тока
- Эксплуатация при температуре окружающего воздуха: -20...+50 °С и Влажности: 30...80 % при температуре +35 °С

Одноканальный ПИД-регулятор ОВЕН ТРМ10-Щ1.У.СР изображен на рисунке 3.

Терморегулятор ОВЕН ТРМ10-Щ1.У.СР предназначен для измерения температуры, регулирования по ПИД закону.

Функциональные возможности

- ОВЕН ТРМ10-Щ1.У.СР рекомендуется применять для управления объектами, обладающими повышенной инерционностью, где обычное регулирование не обеспечивает необходимую точность.



Рисунок 3 - ОВЕН ТРМ10-Щ1.У.СР

- ОВЕН ТРМ10-Щ1.У.СР может управлять:
  - процессом нагрева
  - процессом охлаждения объекта

- Класс точности регуляторов ТРМ10-Щ1.У.СР: 0,5/0,25.
- Назначение: общее
- Тип выхода: симистор/реле

ТРМ210 ПИД-регулятор с универсальным входом и RS-485 изображен на рисунке 4.

Создан для точного поддержания необходимых и заранее известных параметров во всевозможных технологических процессах, в которых требуется контроль температуры.



Корпус Н2



Корпус Н

Рисунок 4 - ТРМ210 с RS-485

- Класс точности регулятора 0,5/0,25.
- Прибор выпускается в 4-х типах корпусов: настенном Н, щитовых Щ1, Щ2 и новом эргономичном корпусе Н2.
- Два выхода в любых комбинациях: электромагнитное реле / оптосимистор / оптотранзистор / 4...20 мА / 0..10 В / внешнее твердотельное реле.
- Встроенный интерфейс RS-485
- Напряжение питания: 90...245 В переменного тока
- Частота напряжения питания: 47...63 Гц

- Рекомендуется использование при следующих условиях окружающей среды:
  - Температура воздуха, взаимодействующего с прибором: +1...+50 °С
  - Атмосферное давление: 0,85...1,05 кгс/см<sup>2</sup>
  - Относительная влажность воздуха (при 35 °С): 30...80 %

Для лучшего понимания вопроса сравним цены представленных образцов в таблице 1.

Таблица 1 - Примеры ПИД-регуляторов температуры.

Наименование	Цена, руб.
Maxwell PMA-49-R-2-96-N-N	3000
ОВЕН ТРМ500	3600
ОВЕН ТРМ10-Щ1.У.СР	5724
ОВЕН ТРМ210 с интерфейсом RS-485	7452

Выводы по разделу

Исходя из совокупности вышеизложенных начальных данных, а также обзору аналогов, существующих в данное время на рынке, мы можем заметить, что цена представленных образцов начинается от 3000 и заканчивается 7452. Важно отметить, что есть и более дорогие образцы.

Иными словами, для нашего случая ПИД-регулятора температуры данный диапазон цен показался наиболее адекватным и мы решили ограничиться ими. Как уже упоминалось ранее, но все же рассмотрим их поподробнее: интерфейс RS-485 и встроенный Wi-fi модуль поднимает цену регулятора, однако для нас они не сильно важны. Для нас необходимо чтобы ПИД-регулятор поддерживал заданную температуру.

Таким образом, когда мы поняли, что для нас необходимо и посмотрели существующие устройства, мы можем спроектировать свой вариант ПИД-регулятора.



## 2. Аппаратная часть

### 2.1 Создание структурной схемы

В соответствии с основными параметрами, которым должна удовлетворять модель ПИД-регулятора температуры, а именно:

- Источник питания 220В, 50 Гц
- Отображение температуры окружающего воздуха
- Автоматическое регулирование функции нагрева по встроенному алгоритму

Для управления ПИД-регулятором была выбрана плата компании Arduino, по причине своей низкой стоимости, а также простотой своего использования и из-за того, что подходит для управления составляющими компонентами.

Для того, чтобы улучшить наш ПИД-регулятор, было решено добавить к нему схему для определения уровня нуля питающей сети, чтобы не было ситуаций, когда регулятор дает указание для воздействия, но оно не может произойти, т.к. в данный момент у нас 0.

Структурная схема была разработана с использованием онлайн-программы для 3D-моделирования Tinkercad, которая позволяет создать и проверить работоспособность будущего проекта, а также запрограммировать в ходе этого процесса выбранный контроллер.

В разработанной структурной схеме имеется в наличии несколько функциональных узлов:

- Микроконтроллер – основа всей системы управления.
- Датчик контроля температуры.
- Блок питания – необходим для питания микроконтроллера, а для этого преобразующий сетевое ~220 В напряжения в необходимое постоянное 9 В.
- Дисплей – для вывода измеренной температуры.

- Датчик напряжения – для детектирования уровня нуля сетевого напряжения.
- Оптопара – для защиты микроконтроллера от напряжения сети, а также для передачи сигнала от микроконтроллера на симистор.
- Симистор – для подачи напряжения на нагреватель.
- Лампа накаливания – используется в качестве нагревателя окружающего воздуха.

На рисунке 5 представлена структурная схема ПИД-регулятора температуры.

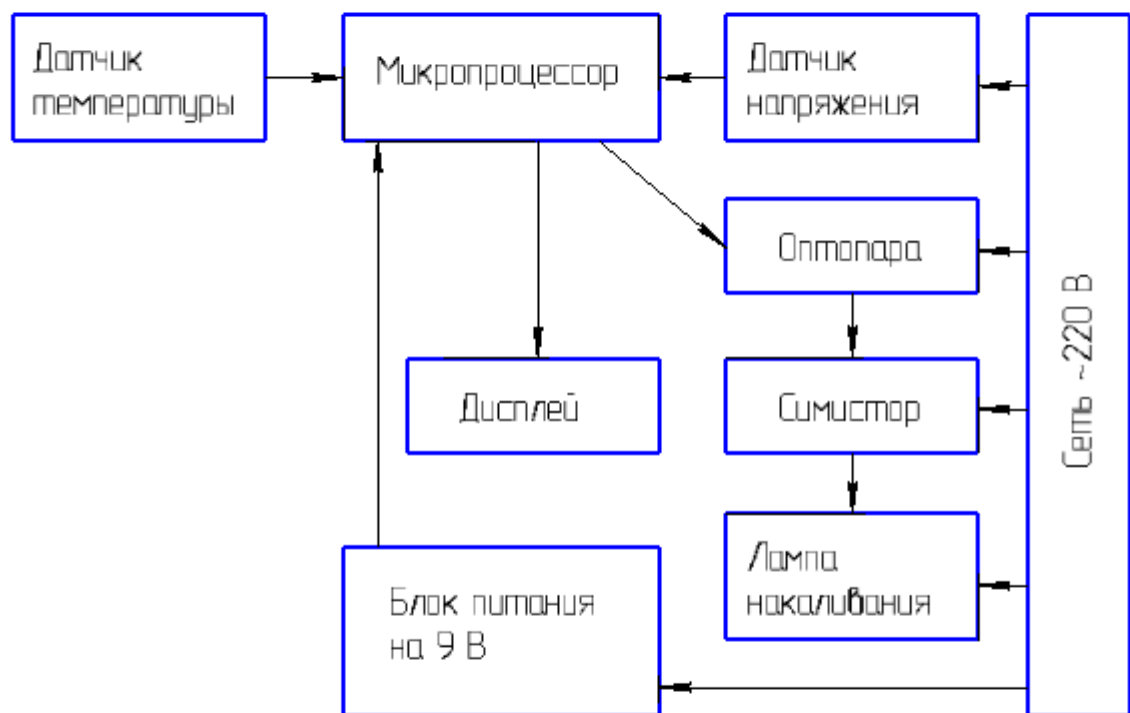


Рисунок 5 - Структурная схема ПИД-регулятора температуры

## **2.2 Проектирование электрической принципиальной схемы**

Данная схема позволит нам увидеть весь проект в подробностях, полное подключение компонентов, а также взаимосвязь между каждым элементом в условных графических обозначениях и буквенно-цифровых подписях.

Она прекрасно подойдет для того, чтобы решить возможные неполадки при неправильном подключении, а также для наладки и контроля. Позволит увидеть все провода и детали конструкции, а не как в реальности – когда они могут перекрывать друг друга. Схема наглядно помогает разобраться в принципе работы устройства.

Была реализована через программу Компас в соответствии действующим ГОСТ и положением об оформлении чертежей, а именно схемы электрической принципиальной. Моделирование и проверка на работоспособность была произведена в Tinkercad, также как и структурная схема.

На рисунке 6 изображена принципиальная схема ПИД-регулятора температуры.

1. Разъем USB DD1 соединить с целью питания +9В X1

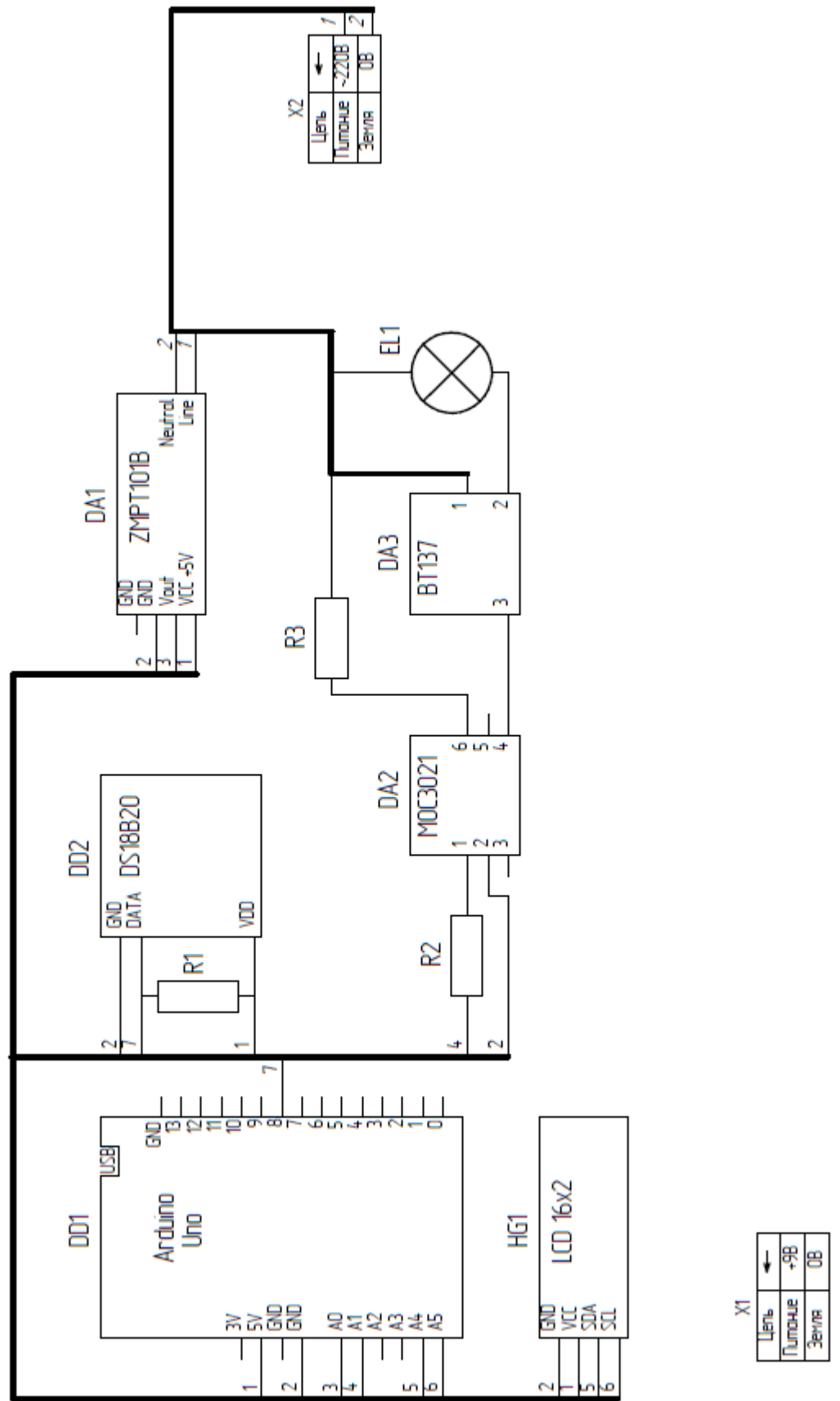


Рисунок 6 – Принципиальная схема ПИД-регулятора

## 2.3 Подбор нужных элементов

Проектируемое устройство ПИД-регулятор температуры содержит в своем составе следующие составляющие:

- Микроконтроллер – основная часть всего устройства, как управляющий компонент.
- Датчик контроля температуры.
- Блок питания – необходим для питания микроконтроллера, а для этого преобразующий сетевое ~220 В напряжения в необходимое постоянное 9 В.
- Дисплей – для вывода измеренной температуры.
- Датчик напряжения – для детектирования уровня нуля сетевого напряжения.
- Оптопара – для защиты микроконтроллера от напряжения сети, а также для передачи сигнала от микроконтроллера на симистор.
- Симистор – для подачи напряжения на нагреватель.
- Лампа накаливания – используется в качестве нагревателя окружающего воздуха.
- Макетная плата – для удобства сборки и подключения компонентов.

В качестве системы управления ПИД-регулятором температуры была выбрана схема Arduino с микроконтроллером ATmega328 (рисунок 7). Она позволяет реализовать все необходимые для успешного выполнения проекта функции, а также является достаточно доступной, имеет низкую стоимость при прочих равных параметрах и для нее есть большое количество бесплатных уроков и скетчей.

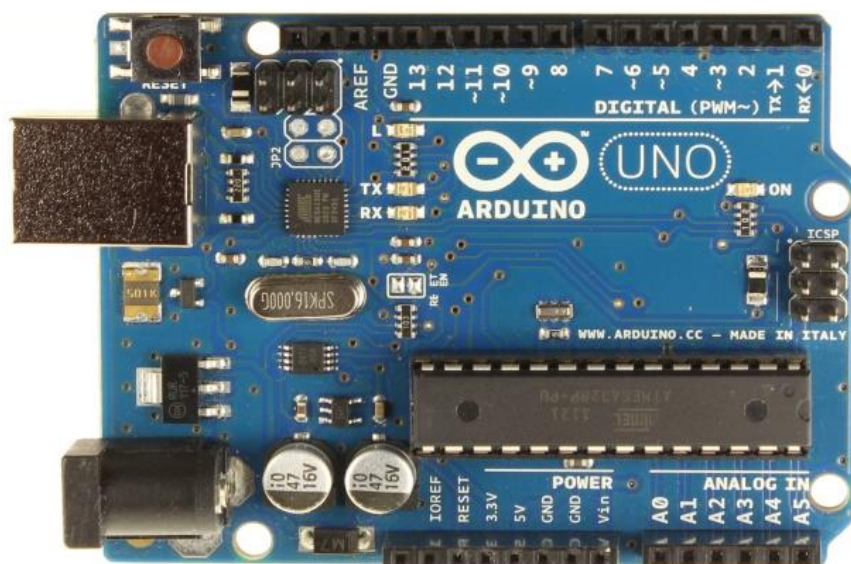


Рисунок 7 – Плата Arduino Uno

Что же из себя представляет данная плата? Это совокупность элементов, основанная на микроконтроллере ATmega328. Иными словами – это плата, которая является «передатчиком» между пользователем и контроллером. Она предназначена для создания разных проектов, как учебных, так и более профессиональных. Благодаря удобным штырькам-разъемам позволяет присоединять различные устройства необходимые в конкретный момент.

На сегодняшний день является одной из самых современных, имеет большое число как встроенных, так и написанных другими пользователями библиотек на многие случаи.

Была разработана в Италии, в качестве более дешевого аналога для обучения программированию. Существует множество копий, которые появились из-за растущего спроса с немного другими характеристиками в отличие от оригинальной платы.

Характеристики оригинальной платы:

- Микроконтроллер: ATmega328
- Рабочее напряжение: 5 В
- Питающее напряжение (оптимальное): 7-12 В
- Питающее напряжение (максимальное): 6-20 В

- Максимальный ток через разъемы 5 В: 40 мА
- Максимальный ток для вывода 3.3 В: 50 мА
- Флеш-память: 32 Кб (АТmega328) из которых 0.5 Кб используются загрузчиком
- ОЗУ: 2 Кб (АТmega328)
- EEPROM: 1 Кб (АТmega328)
- Тактовая частота: 16 МГц

Имеются 20 выводов из которых 14 являются цифровыми, а 6 - аналоговыми, которые вместе с их назначением и адресацией приведены в таблице 2.

Таблица 2 – Назначение выводов платы Arduino

Пин Arduino		Адресация пина в скетче	Альтернативное назначение	ШИМ
Цифровые	Пин №0	0	RX	
	Пин №1	1	TX	
	Пин №2	2	Входы для внешних прерываний	
	Пин №3	3		да
	Пин №4	4		
	Пин №5	5		да
	Пин №6	6		да
	Пин №7	7		
	Пин №8	8		
	Пин №9	9		да
	Пин №10	10	SPI (SS)	да
	Пин №11	11	SPI (MOSI)	да
	Пин №12	12	SPI (MISO)	
	Пин №13	13	SPI (CSK) + светодиод	

Продолжение таблицы 2

Пин Arduino	Адресация пина в скетче	Альтернативное назначение	ШИМ	
Аналоговые	Пин №A0	A0		
	Пин №A1	A1		
	Пин №A2	A2		
	Пин №A3	A3		
	Пин №A4	A4	I2C (SDA)	
	Пин №A5	A5	I2C (SCL)	

Теперь, когда мы рассмотрели небольшое описание того, что же такое Arduino, ознакомимся что и где располагается на самой плате более подробно(рис.8)

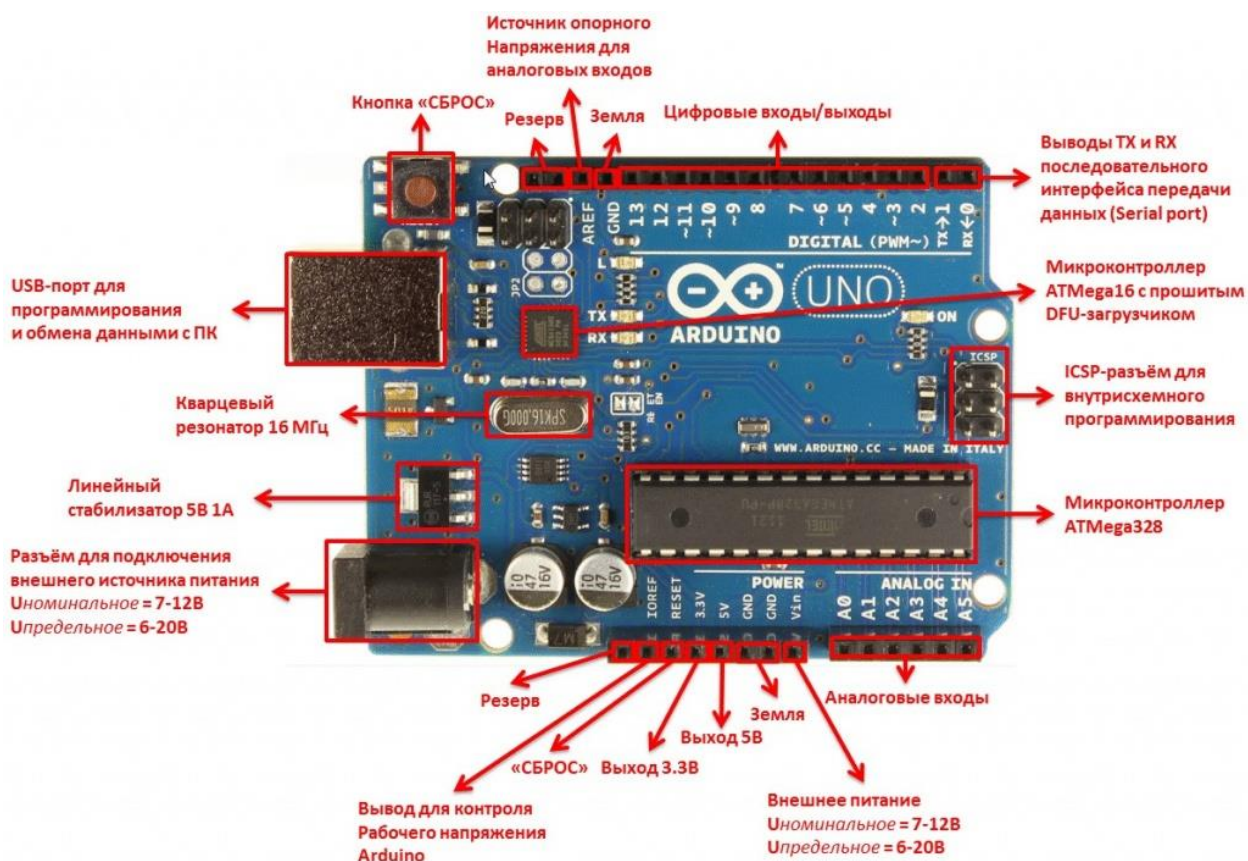


Рисунок 8 – Расположение элементов на плате Arduino



## Соединение с ПК и программирование

Вверху слева на рис.8 расположен USB-порт. Он служит для обмена информацией между компьютером или иным устройством, например ноутбуком и т.д. и микроконтроллером ATmega328, а также записью скетчей в него же.

Посредником между ПК и микроконтроллером ATmega328 является отдельно встроенный ATmega16. Он нужен для того, чтобы ПК мог определять Arduino как виртуальный COM-порт при подключении. Во время обмена информацией происходит мигание светодиодов, располагающихся справа от ATmega16 и левее надписи Arduino на плате.

Запись скетча происходит при нажатии всего 1 кнопки в среде разработки Arduino IDE (рис.9), которая является и бесплатной и доступной как на официальном сайте, так и на многочисленных форумах.

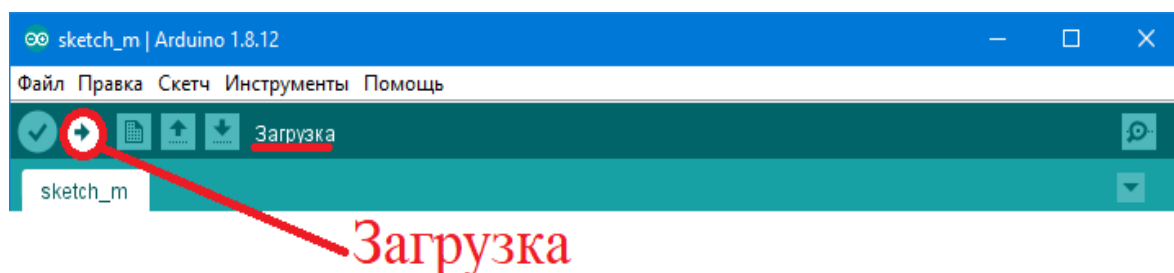


Рисунок 9 - Среда разработки Arduino IDE

## Система питания

Для успешного функционирования Arduino, ей как и другому устройству, необходимо питание. Есть несколько вариантов как это можно сделать:

- USB-порт через шнур, соединенный с работающим ПК
- Блок питания на 7-12В через разъем «внешний источник питания», расположенный слева снизу на рис.8.

- Непосредственно на разъем Vin 7-12В, расположенный в группе контактов снизу платы на рис.8, а «минус» источника соединим с GND платы

Плата может работать при питающем напряжении от 6 В до 20 В. Оптимальны считается 7-12В. Если брать ниже границы оптимального, то работа может быть нестабильной, а вывод 5В будет менее своего значения. Если же подавать больше 12, то наиболее вероятен перегрев регулятора и выход его из строя, что также повредит плату.

#### Выводы питания

- VIN. Предназначен для питания от внешнего источника при отсутствии 5 В через USB или другого варианта описанного выше.
- 5V. Используется для питания микроконтроллера и элементов на плате, питается от вывода VIN через регулятор напряжения / от разъема USB / другого регулируемого источника напряжения 5 В.
- 3V3. Напряжение на выводе 3.3 В - генерируется встроенным регулятором на плате. Максимальное значение тока 50 мА.
- GND. Для заземления, соединены между собой.

#### Токовая защита разъема USB

В Arduino имеется встроенный предохранитель (автомат), который предназначен для защиты порта USB компьютера от токов короткого замыкания. Срабатывает при значении тока более 500 мА через USB порт и соответственно разрывает цепь до восстановления нормальных значений.

#### Присоединение элементов

Осуществляется через контакты путем их подключения к одному из выводов, которые необходимы для конкретного устройства. Как правило, минимальное число задействованных пинов при этом 2: питание и

заземление. Для датчиков 3: кроме вышеуказанных добавляется цифровой или аналоговый пин данных.

Во избежание повреждения устройства следует заранее рассчитывать токи, чтобы они находились в разрешенных пределах.

Важно помнить: максимальное значение тока на один вывод 40 мА, рекомендуемое – 20 мА. Также не следует превышать предельное суммарное значение 200 мА.

### Arduino LCD 1602 I2C дисплей

Жидкокристаллический дисплей (Liquid Crystal Display) LCD 1602 подходит, когда необходимо выводить символы в строку в разных проектах (рисунок 10). Отличается невысокой стоимостью, имеется большое число вариантов с разными версиями подсветки, часто используется пользователями, благодаря чему вы с большой вероятностью найдете множество вариантов библиотек готового кода, подходящих именно вашей версии дисплея Ардуино.

Однако существенным недостатком данного экрана является следующая печальная действительность: дисплей имеет 16 цифровых выводов, из которых безусловно необходимыми являются минимум 6.

Вследствие данного факта использование такого LCD экрана в том виде, в каком он выпускается является не совсем удобным. Без модуля i2c, который служит модификацией для того, чтобы избавить нас от существенных ограничений при подключении его к плате Arduino. Проблема в том, что в большинстве проектов с большой долей вероятности контактов вам очевидно будет не хватать, из-за чего вам придется решать эту случившуюся неприятность путем покупки либо большего размера платы, такой как Arduino Mega – с существенно увеличенным числом контактов или же каким-либо образом уменьшать количество контактов, вполне вероятно и за счет использования дисплея через i2c.

## Краткое описание пинов LCD 1602



Рисунок 10 - Arduino LCD 1602 I2C дисплей

Как мы видим на рисунке 10 - каждый из выводов имеет свое конкретное назначение, а именно:

1. Вывод №1 VSS (Ground) – служит для заземления устройства;
2. Вывод №2 VDD (+ve) – предназначен для питания дисплея напряжением 5 В;
3. Вывод №3 VE (Contrast Voltage) – служит для задания необходимой контрастности монитора;
4. Вывод №4 Register Select - определен для команд, данных;
5. Вывод №5 Read / Write - рассчитан для записи и чтения поступающих данных;

6. Вывод №6 Enable – уготован для включения/разрешения;  
Выводы № 7-14 Data – предназначены для линии данных;  
15. Вывод №15 Backlight Anode (+ve) – рассчитан для «плюс» подсветки;  
16. Вывод №16 Backlight Cathode - служит для «минус» подсветки.

Технические характеристики дисплея:

- При возможности загрузки символов – отображение символьное;
- Подсветка производится светодиодами;
- Контроллер HD44780;
- Питающее напряжение 5В;
- Размер 16x2 символов;
- Охват рабочих температур от -20С до +70С, охват температур содержания от -30С до +80 С;
- Ракурс обзора 180 градусов.

Один из простейших вариантов схемы I2C может включать одно головное устройство (наиболее вероятно это микроконтроллер Arduino) и несколько вспомогательных (например, дисплей LCD). Каждое приспособление имеет адрес в размере от 7 до 127. Двух устройств с одинаковым адресом в одной схеме не представлено возможным.

Плата Arduino поддерживает i2c на аппаратном уровне. Вы можете применять пины A4 и A5 для подсоединения устройств по вышеназванному протоколу.

В работе I2C можно отобразить некоторое число достоинств:

- Для работы требуется всего 2 линии – SDA (линия данных) и SCL (линия синхронизации).
- Присоединение большого количества головных устройств.
- Сокращение времени разработки.
- Для управления всем набором устройств необходим лишь один микроконтроллер.

- Вероятное количество присоединяемых микросхем к одной шине лимитируется только максимальной емкостью.
- Высокая степень сохранности информации из-за специального фильтра заглушающего всплески, интегрированного в схемы.
- Простая операция по диагностике возможных неполадок, быстрое исправление неполадок.
- Шина уже встроена в саму плату Arduino, поэтому не нужно разрабатывать дополнительно шинный интерфейс.

Несовершенства устройства:

- Присутствует емкостное лимитирование на линии – 400 пФ.
- Усложненное программирование контроллера I2C, если на шине находится некоторое количество различных приборов.
- При превышении числа устройств случаются трудности локализации сбоя, если одно из них ошибочно утверждает состояние низкого уровня.

Модуль i2c для LCD 1602 Arduino (рис. 11)

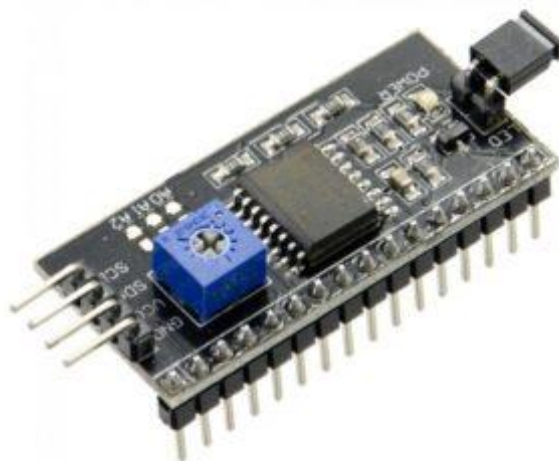


Рисунок 11 - Модуль i2c для LCD 1602 Arduino

Наиболее скоростной и подходящий способ использования i2c дисплея в Arduino – это покупка готового дисплея с интегрированной поддержкой протокола.

Но такие версии дисплеев не особо распространены и отличаются большей стоимостью по сравнению с вариантом без поддержки. К тому же количество разнообразных стандартизированных дисплеев довольно высоко и продолжает выпускаться.

Вследствие чего наиболее оптимальным в плане доступности, а также популярности является приобретение и применение отдельного I2C модуля – переходника, который описан ниже:

С одной стороны переходника мы можем увидеть выводы i2c – земля, питание и 2 для передачи данных (рис.12). С другой – расположены разъемы для внешнего питания. И, конечно, на самом устройстве имеется множество ножек, с помощью которых модуль присоединяется к типовым выводам экрана.

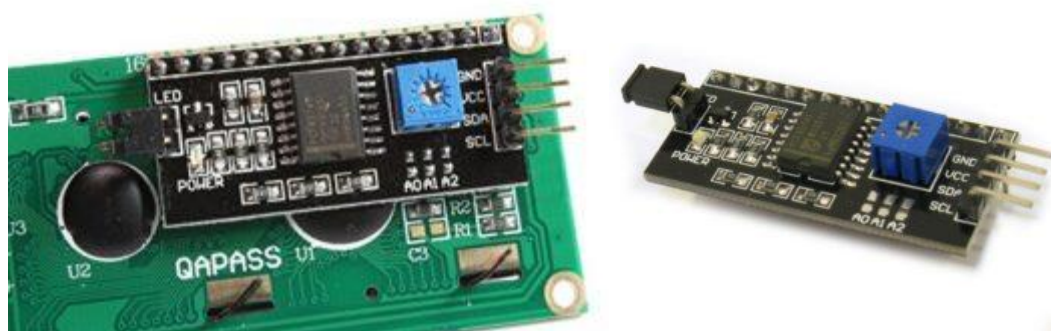


Рисунок 12 - I2C модуль – переходник

### Подключение ЖК экрана к Arduino по I2C

Жидкокристаллический монитор с поддержкой i2c присоединяется к плате благодаря четырем проводам – два необходимы для данных, два используются для питания (рис.13).

- SCL соединяется с пином A5.
- SDA соединяется с пином A4.
- Вывод GND соединяется с GND, расположенном на плате.
- Вывод VCC – соответственно на 5V.

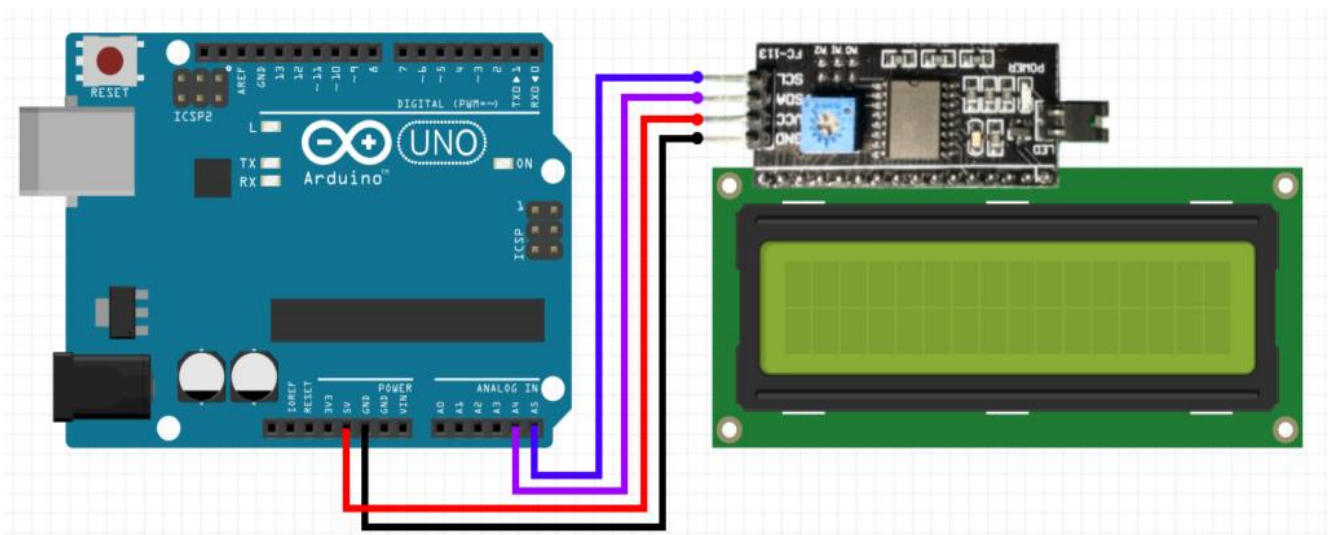


Рисунок 13 – Подключение ЖК экрана к Arduino по I2C

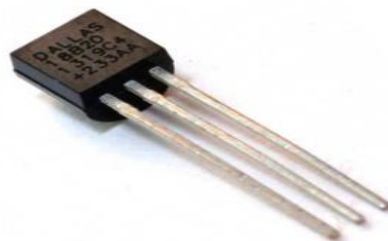
Библиотеки для взаимодействия с i2c LCD дисплеем

Для начала использования Arduino с LCD 1602 по шине I2C вам будут необходимы как минимум две библиотеки:

- Библиотека `Wire.h` для работы с I2C уже находится в типовой программе Arduino IDE.
- Библиотека `LiquidCrystal_I2C.h`, включающая в себя значительное многообразие команд для управления дисплеем по шине I2C и дает возможность сделать код программы проще и короче. Для этого необходимо дополнительно подключить библиотеку. После чего для подключения дисплея нужно будет установить вышеописанную библиотеку `LiquidCrystal_I2C.h`

Датчик температуры DS18B20

Интегральный датчик температуры DS18B20 с интерфейсом 1 – Wire (рис.14)





## Рисунок 14 - DS18B20 с интерфейсом 1 – Wire

Исключительны качества DS18B20:

- Для интерфейса 1-Wire, использующего один провод будет довольно одного порта для связи с микроконтроллером.
- У каждого произведенного датчика есть свой только ему присущий серийный код длиной 64 разряда.
- С большой долей вероятности допустимо присоединение нескольких датчиков через одну линию связи.
- Для использования не нужны внешние компоненты.
- Допустимо получение питающего напряжения напрямую от линии связи. Питающее напряжение в диапазоне 3,0 В ... 5,5 В.
- Размах измерения температуры -55 ... +125 °С.
- Погрешность не превышает 0,5 °С в диапазоне -10 ... +85 °С.
- Разрешение преобразования задается пользователем. Обычно находится в пределах 9 ... 12 бит..
- Время необходимое для измерения, как правило, не превышает 750 мс, при предельно доступном разрешении 12 бит.
- Допускается программирования параметров тревожного сигнала.
- Тревожный сигнал передает данные об адресе датчика, у которого температуры вышла за заданные пределы.
- Совместимость программного обеспечения с DS1822.
- Крайне широкие области применения.

Назначение выводов представлено в таблице 3.

Таблица 3 – Назначение выводов

8-PIN SOIC	TO-92	СИГНАЛ	ОПИСАНИЕ
5	1	GND	Вывод земля
4	2	DQ	Предназначен для вывода сигнала данных (входа/выход). Выход представляет собой открытый коллектор интерфейса 1-Wire. К тому же через него производится питание в режиме ”паразитное

			питание”.
--	--	--	-----------

Продолжение таблицы 3

8-PIN SOIC	TO-92	СИГНАЛ	ОПИСАНИЕ
3	3	V <sub>DD</sub>	Служит в качестве вывода для внешнего питания. В режиме ”паразитного питания” должен быть подключен к земле.

На рисунке 15 показаны назначение выводов DS18B20

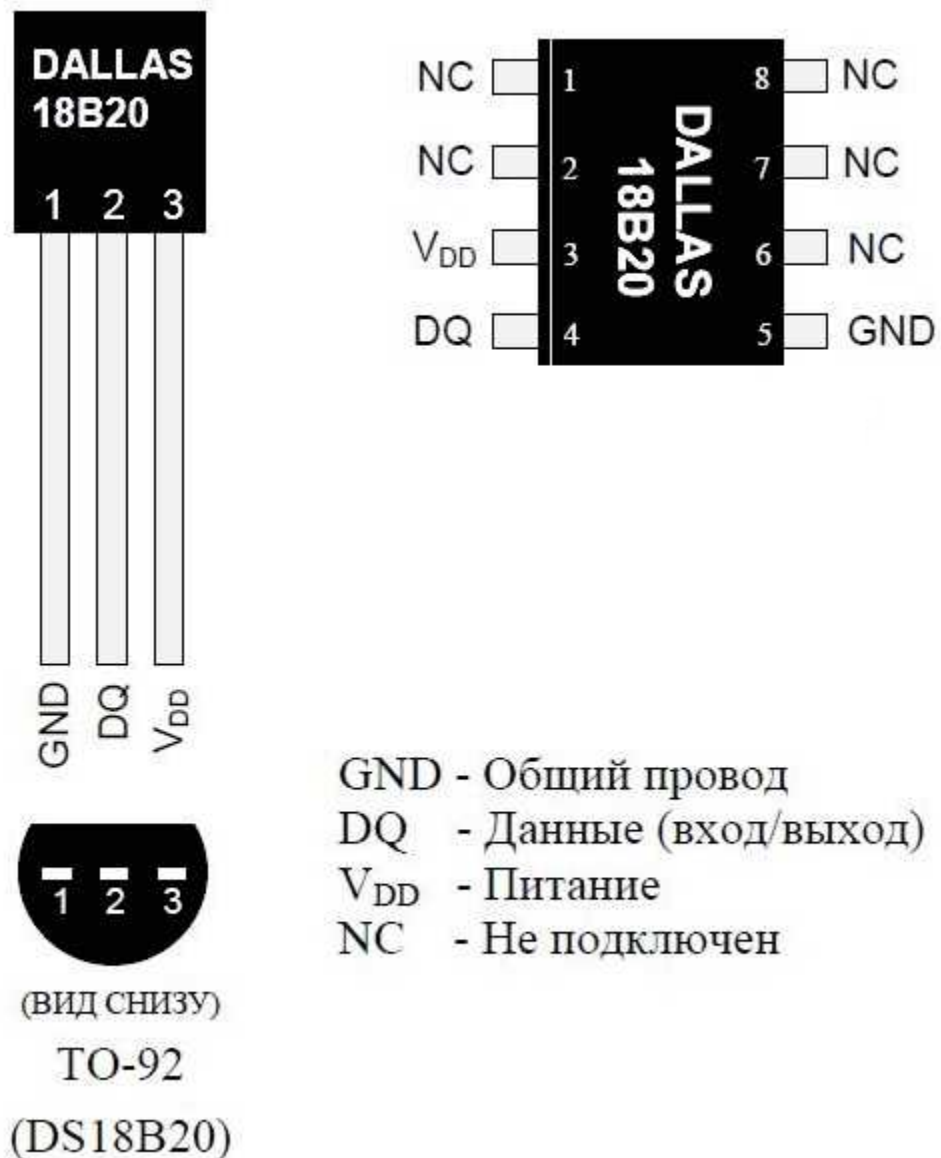


Рисунок 15 – Назначение выводов DS18B20

Резистор 4.7 кОм (рис.16)

- Предназначен в данном проекте для подключения DS18B20
- Номинальное сопротивление : 4.7 кОм
- Точность, %: 5
- Номинальная мощность, Вт: 0.5
- Максимальное рабочее напряжение, В: 350
- Рабочая температура, С: -55...155
- Монтаж: в отв.
- Длина корпуса L, мм: 9
- Ширина (диаметр) корпуса W(D), мм: 3.2
- Вес, г: 0.26



Рисунок 16 – Резистор 4.7 кОм

Датчик переменного напряжения для Arduino ZMPT101B (рис.17)

ZMPT101B – модуль на основе одноимённого понижающего трансформатора напряжения, позволяющий измерять напряжение переменного тока от 195 до 250 В. Для приведения показаний в более-менее читаемый микроконтроллером вид используется операционный усилитель, установленный на плату.

Всё, что нужно для начала работы – подключить питание 5В от Arduino, высокое напряжение на вход трансформатора и аналоговый выход к любому аналоговому пину Arduino. Ток, потребляемый из сети – до 2 мА.

Основные преимущества

- Изолированное преобразование без риска появления высокого напряжения в выходной цепи

- Высокая точность благодаря прецизионному операционному усилителю
- Низкий ток потребления



Рисунок 17 - Датчик переменного напряжения для Arduino ZMPT101B

Оптопара МОС3021М (рис.18)

Предназначена для приема и передачи сигнала без прямого контакта. Служит для защиты Arduino от чрезмерного уровня напряжения питающей сети.

- |                                       |              |
|---------------------------------------|--------------|
| • Характер выхода:                    | фототиристор |
| • Напряжение пробоя изоляции, кВ:     | 7.5          |
| • Предельный прямой ток, мА:          | 60           |
| • Предельное напряжение на выходе, В: | 250          |
| • Время включения/выключения, мкс:    | 10           |
| • Тип корпуса:                        | dip6         |
| • Вес, г:                             | 0.8          |

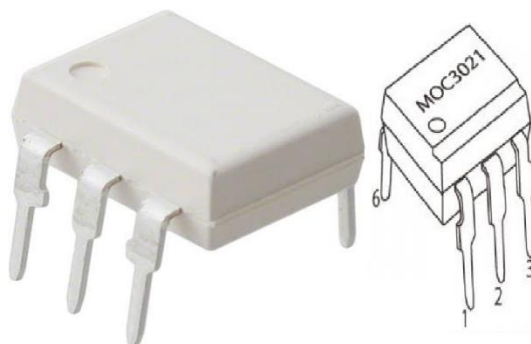


Рисунок 18 – Оптопара МОС3021М

### Симистор ВТ137-600Е (рис.19)

- Симистор (симметричный тиристор)
- Предельное значение импульса напряжения в закрытом состоянии, В: 600
- Предельное среднее за значение тока в открытом состоянии, А: 8
- Постоянный ток управления, требуемый для начала работы тиристора, мА: 2.5
- Удерживающий ток, необходимый для поддержания открытого состояния тиристора, мА: 3
- Предельный импульсный ток в открытом состоянии, А: 8
- Напряжение включения (открывания) при 25оС, В: 0.7
- Предельное значение напряжения в открытом состоянии (падение напряжения), В: 1.3
- Максимальная скорость увеличения напряжения в закрытом состоянии  $dU_{зс.}/dt$ , В/мкс: 50
- Максимальная скорость увеличения тока в открытом состоянии  $dI/dt$ , А/мкс: 50
- Необходимое для начала работы время  $t_{вкл.}$ , мкс: 2
- Диапазон температуры для работы, С: -40...125



Рисунок 19 - Симистор ВТ137-600Е

### Резисторы 330 Ом и 1 кОм (рис.20)

- Для подключения МОС3021М

- Резисторы используются при работе в цепях постоянного, переменного и импульсного тока.
- Номинальная мощность: 0.25 Вт, 0.5 Вт, 1 Вт, 2 Вт
- Точность: 5% (J)
- Диапазон рабочих температур: -55 ...+125°C



Рисунок 20 - Резисторы 330 Ом и 1 кОм

Также была рассчитана экономическая составляющая компонентов ПИД-регулятора (таблица 4).

Таблица 4 – Стоимость компонентов для создания ПИД-регулятора температуры

Наименование	Цена, руб.
Устройство Arduino Uno	170
Arduino LCD 1602 I2C дисплей	280
Датчик температуры DS18B20	180
Резистор 4.7 кОм	4
Датчик переменного напряжения для Arduino ZMPT101B	170
Оптопара МОС3021М	22
Симистор BT137-600E	44
Резисторы 330 Ом и 1 кОм	8
Макетная плата	89

Для лучшего понимания того, как выглядит сборка из подобранных компонентов представлен рисунок 20.

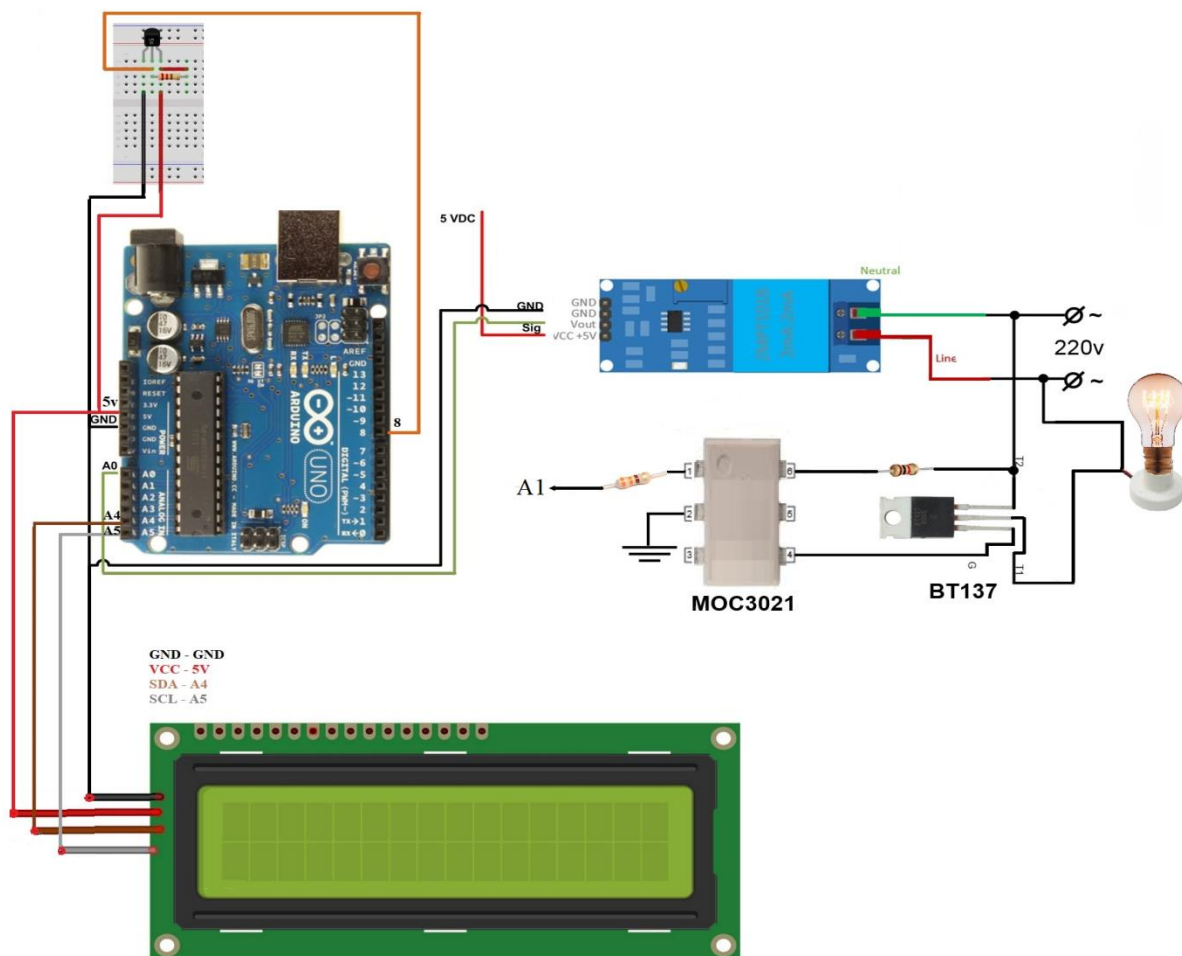


Рисунок 20 – Схема соединения компонентов ПИД-регулятора температуры

Можно увидеть физическую реализацию проекта, а именно:

Устройство Arduino Uno – расположен слева посередине.

Arduino LCD 1602 I2C дисплей – разместился под Arduino.

Датчик температуры DS18B20 – находится выше микроконтроллера на макетной плате.

Резистор 4.7 кОм – там же, где и DS18B20.

Датчик переменного напряжения для Arduino ZMPT101B – сверху по центру.

Оптопара MOC3021M – под датчиком напряжения

Симистор BT137-600E – левее лампы накаливания.

Резисторы 330 Ом и 1 кОм – в верхней части оптопары.

Макетная плата – над Arduino.

## Выводы по разделу

В данном разделе мы разработали в соответствии с основными параметрами, которым должна удовлетворять модель ПИД-регулятора температуры структурную схему устройства. И описали функциональные узлы из которых она состоит.

Затем мы спроектировали на основе имеющихся данных о подключении устройств электрическую принципиальную схему.

После чего, рассмотрев и проанализировав информацию, были подобраны необходимые компоненты для успешного функционирования нашего ПИД-регулятора температуры, а также рассмотрены краткие описания элементов с их возможным подключением и действиями, которые требуются для их работы, такие как: рекомендуемые параметры питания и библиотеки для использования девайсов.



## **3. Программная часть**

### **3.1 Разработка алгоритма работы**

Алгоритм работы – некая последовательность действий, делающая код программы более понятным для ознакомления, а также, что вполне возможно, его редактирования (рис.21).

В создаваемом проекте ПИД-регулятор температуры, который будет выполнять контроль и поддержание уровня температуры на заданном значении, а также проверка сетевого переменного напряжения на значение нуля. В зависимости от того, какая температура была измерена, микроконтроллер по записанным в коде программы действиям будет принимать решение о необходимости нагрева окружающего воздуха, если измеренная температура меньше заданной или не делать ничего – если выше.

Сам микроконтроллер будет питаться от блока питания или же иными способами описанными выше, например от USB-порта компьютера.

Часть, связанная с нагревом, а также детектированием уровня нуля сетевого напряжения соединена непосредственно через провода с переменным напряжением 220 В.

А устройства, которым необходимо постоянное напряжение: дисплей и датчик температуры – подключены к выводу 5В микроконтроллера.

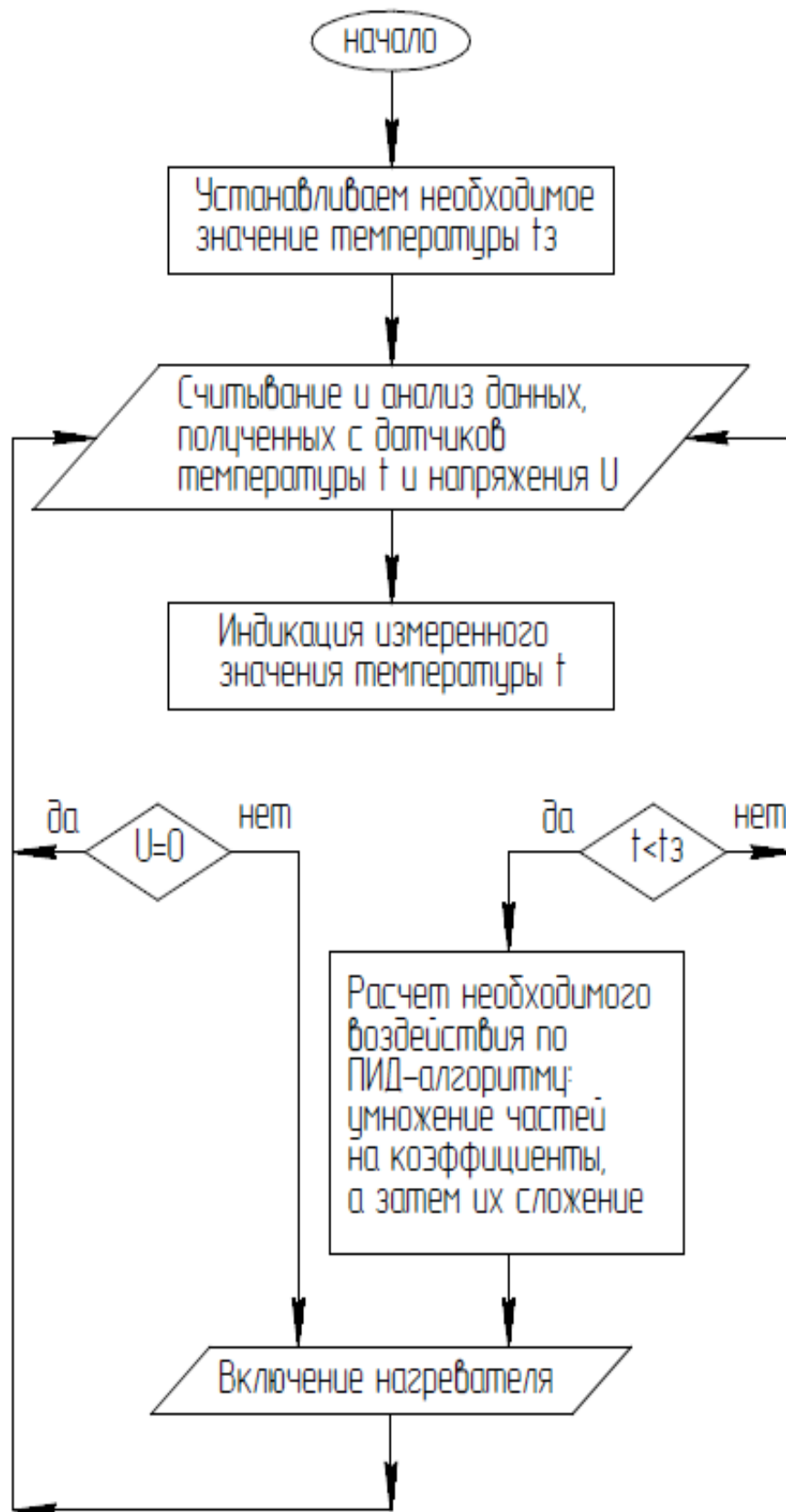
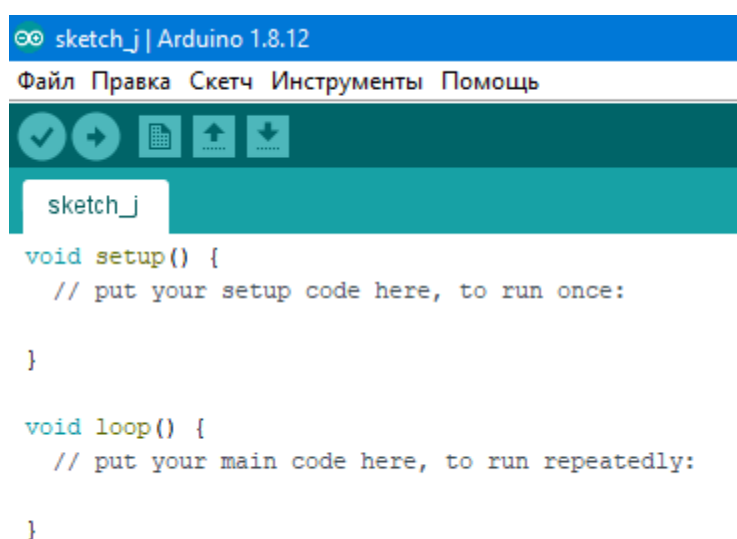


Рисунок 21 – Блок-схема алгоритма

## 3.2 Разработка кода программы

В основе своей при программировании Arduino лежит упрощенный язык C++. Для того, чтобы начать непосредственно писать программы для данного микроконтроллера вам потребуется: специальное программное обеспечение (рис.22), которое можно скачать и установить на ПК или ноутбук с официального сайта или многочисленных форумов; сам микроконтроллер Arduino; шнур USB – для подключения их друг к другу.

Код программы, написанный с использованием данного программного обеспечения носит название «скетч». Для сокращения величины кода создаются библиотеки, многие из которых являются стандартными и встроенными сразу при установке. Также есть большое количество сторонних библиотек, созданных другими пользователями, которые также можно загрузить для использования.



```
sketch_j | Arduino 1.8.12
Файл Правка Скетч Инструменты Помощь
sketch_j
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Рисунок 22 – Среда разработки Arduino

В своем скетче мы используем следующие библиотеки:

- `#include <Wire.h>` - предназначена для взаимодействия с устройствами по шине I2C, которые используются в данном проекте;
- `#include <LiquidCrystal_I2C.h>` - служит для подключения и управления нашим дисплеем;

- `#include <DallasTemperature.h>` - необходима для упрощения работы с датчиком температуры.

После того, как в скетче мы написали про подключение библиотек, описанных выше, мы определяем размер и адрес дисплея, а также подключаем его и датчики. Затем прописываем коэффициенты для ПИД-регулирования: пропорционального, дифференциального и интегрального.

После чего мы измеряем температуру воздуха окружающего датчик и выводим ее на экран. После чего происходит ее сравнение с заданным значением.

В это же время у нас происходит проверка сетевого напряжения на уровень нуля, и если измеренная температура ниже заданной и одновременно не ноль по напряжению, то вычисляем разницу температур и отправляем управляющее воздействие, необходимое для того, чтобы значения этих температур стали равными т.е. нагреваем воздух.

Данный процесс происходит с частотой сетевого напряжения 50 Гц, что является оптимальным т.к. процессы, связанные с нагревом, а также охлаждением достаточно инерционные.

Далее приведен пример моделирования и форма напряжения сетевого и на лампе накаливания в программе Tinkercad(рис.23)

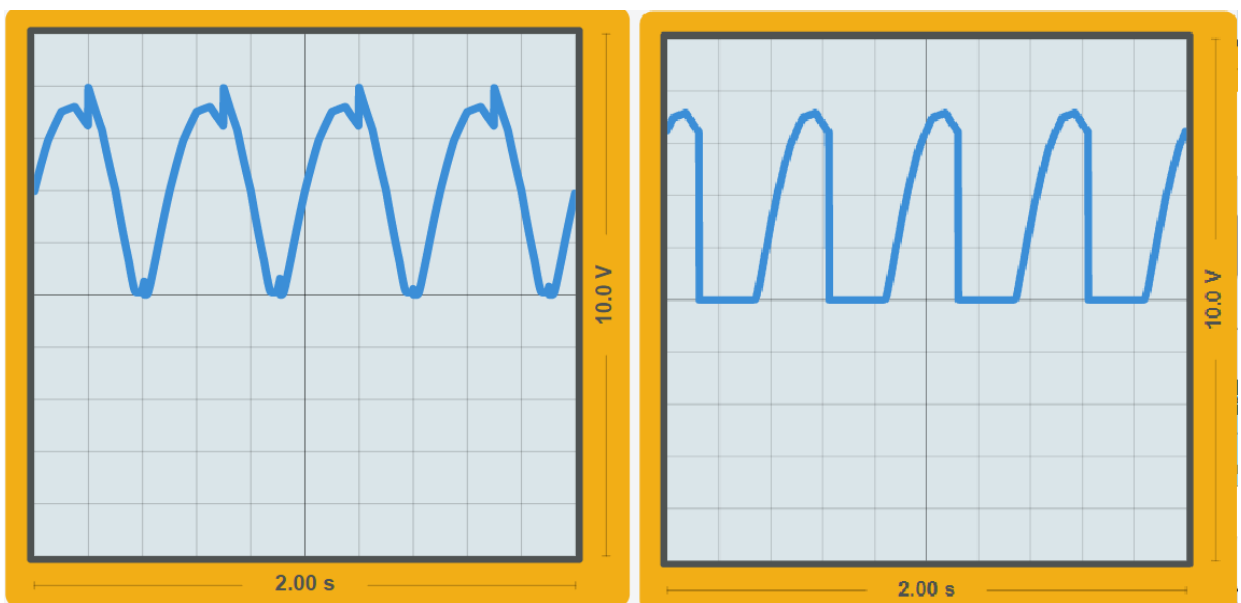


Рисунок 23 – Пример моделирования в Tinkercad.

## Выводы по разделу

Рассмотрены в третьем разделе были алгоритм работы программы, который необходим для более понятного отображения функционирования скетча. Иными словами, что будет, если случится то или иное событие, отраженное в алгоритме.

После чего, был разработан скетч – код программы устройства и рассмотрен процесс подключения как самого микроконтроллера – для чего используется среда Arduino, так и сопутствующие элементы – такие как: библиотеки для дисплея, а также датчиков. Затем описана процедура действий ПИД-регулятора температуры, а именно: что должно произойти для нагрева окружающего воздуха, а также процессы попутно происходящие в микроконтроллере для этого.

## Заключение

При создании и написании данной выпускной квалификационной работы были проанализированы источники по теме, приведенные в списке литературы.

В данной ВКР были выполнены поставленные цели и задачи. Сформулирована и обозначена актуальность. Произведен обзор существующих вариантов на рынке. Создан ПИД-регулятор температуры, включающий в себя датчик температуры, а также элементы для детектирования уровня нуля сетевого напряжения и для управления самим нагревателем. Создана структурная схема. Спроектирована электрическая принципиальная схема.

Подобраны нужные элементы, такие как: устройство Arduino Uno, Arduino LCD 1602 I2C дисплей, датчик температуры DS18B20, резисторы 4.7 кОм, 330 Ом и 1 кОм, датчик переменного напряжения для Arduino ZMPT101B, оптопара МОС3021М, симистор BT137-600E, макетная плата.

Написан код программы для устройства. Создана и отлажена модель ПИД-регулятора.

Итоговый вариант получился почти вдвое дешевле по сравнению с аналогами.

Также, по моему мнению, вполне возможно добавление к функционалу созданного устройства кнопки для установления заданной температуры не программным методом.

Со временем не исключено усложнение устройства и расширение списка его функций. Например, не только нагрев, но и охлаждение воздуха путем добавления к компонентам охлаждающего устройства и, соответственно, отображение этих изменений в коде программы и алгоритме работы.



## Список используемой литературы

1. <https://www.chipdip.ru/product0/44435> [Электронный ресурс]
2. <https://doc.arduino.ua/ru/hardware/Uno> [Электронный ресурс]
3. <https://insat.ru/products/?category=1070> [Электронный ресурс]
4. <http://earchive.tpu.ru/bitstream/11683/28358/1/TPU205096.pdf>  
[Электронный ресурс]
5. <https://webhamster.ru/mytrashare/index/mtb0/1542098495zu6xt8uxnj>  
[Электронный ресурс]
6. [http://robofob.ru/materials/articles/pages/Karpov\\_mobline1.pdf](http://robofob.ru/materials/articles/pages/Karpov_mobline1.pdf)  
[Электронный ресурс]
7. <https://habr.com/ru/post/145991/> [Электронный ресурс]
8. <http://forum.amperka.ru/threads/> [Электронный ресурс]
9. <http://mypractic.ru/> [Электронный ресурс]
10. <https://alexgyver.ru/gyverpid/> [Электронный ресурс]
11. <https://3d-diy.ru/wiki/arduino-platy/arduino-uno/> [Электронный ресурс]
12. <https://arduinomaster.ru/platy-arduino/plata-arduino-uno/>  
[Электронный ресурс]
13. <https://arduinomaster.ru/datchiki-arduino/lcd-i2c-arduino-displey-ekran/>  
[Электронный ресурс]
14. <https://openenergymonitor.org/> [Электронный ресурс]
15. <https://sea.omega.com/tw/prodinfo/temperaturecontrollers.html#:~:text=As%20the%20name%20implies%2C%20a,as%20a%20heater%20or%20fan.> [Электронный ресурс]
16. [https://en.wikipedia.org/wiki/PID\\_controller](https://en.wikipedia.org/wiki/PID_controller) [Электронный ресурс]
17. Mingyi Zhao, Moti Yung CYSARM'19: Proceedings of the 1st ACM Workshop on Workshop on Cyber-Security Arms Race «Secure Zero Detection», 2019, 45 с.



18. Волович Г.И. Схемотехника аналоговых и аналого-цифровых электронных устройств. 3-е изд. стер. / Волович Г.И. – М.: ДМК Пресс, 2015. – 528 с. ил.
19. Zhenran Gao, Lu He, Xiaoguang Yue ICIP '18: Proceedings of the 3rd International Conference on Intelligent Information Processing «Design of PID controller temperature», 2018, 87c
20. Yanying Qi, Zhixian Zhong, Yixin Liu Association for Computing Machinery New York NY United States «PID controller for active magnetic levitation ball system» 2018, 315c
21. Jiayu Liu, Wei Pan, Ruopeng Qu, Meng Xu Association for Computing Machinery New York NY United States « Research on the Application of PID Control with Neural Network and Parameter Adjustment Method of PID Controller» 2018, 76c
22. Jennie Schaeffer, Rikard Lindell TEI '15: Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction «Arduino in Museum Exhibition: Lessons Learned When Working With Design Students Inexperienced in Coding», 2015, 720c
23. Patricia Mellodge, Ingrid Russell ITiCSE '13: Proceedings of the 18th ACM conference on Innovation and technology in computer science education «Using the arduino platform to enhance student learning experiences». 2013, 338c
24. Kayla DesPortes, Betsy DiSalvo ICER '19: Proceedings of the 2019 ACM Conference on International Computing Education Research «Trials and Tribulations of Novices Working with the Arduino» 2019, 227c

## Приложение А

### Код программы

```
#include <Wire.h> // Подключаем библиотеку Wire
#include <LiquidCrystal_I2C.h> // Подключаем библиотеку LiquidCrystal_I2C
#include <DallasTemperature.h> // Подключаем библиотеку DallasTemperature
#define DS18B20 8 // Указываем, к какому выводу подключена DQ

byte simvol[8] = {B11100,B10100,B11100,B00000,B00000,B00000,B00000,B00000,}; //
Символ градуса

LiquidCrystal_I2C lcd(0x27,16,2); // Задаем адрес и размер дисплея
OneWire oneWire(DS18B20);
DallasTemperature sensors(&oneWire);
// П
float up; // пропорциональная составляющая
float kp = 2.2; // коэффициент пропорциональной составляющей
// И
float ui = 0; // интегральная составляющая
float ki = 0; // коэффициент интегральной составляющей
int iSum = 0; // сам интеграл (сумма)
float k_down_i = 0.5; // коэффициент, уменьшающий интегральную
составляющую на каждой итерации

// Д
float ud; // дифференциальная составляющая
float dif=0.05;
float kd = 100; // коэффициент дифференциальной составляющей
int oldErr; // значение ошибки, полученной на предыдущей
итерации
unsigned long time = 0; // время получения текущей ошибки
unsigned long pastTime = 0; // время получения предыдущей ошибки
unsigned long time1 = 0; // время получения текущей ошибки
unsigned long lastTime1 = 0;
unsigned long time2 = 0; // время получения текущей ошибки
unsigned long lastTime2 = 0;
unsigned long time3 = 0; // время получения текущей ошибки
unsigned long lastTime3 = 0;
int maxSpeed = 19; // максимальная скорость
float err;
int pause = 20; // время задержки
const tzad = 28; // заданная температура
void setup() {
sensors.begin(); // Запуск библиотеки, по умолчанию 9 бит, то есть
кратность 0.5 градуса
lcd.init(); // Инициализация lcd
lcd.backlight(); // Включаем подсветку
lcd.setCursor(2,0); // Устанавливаем курсор на 1 строку, ячейка 2
lcd.print("TEMP"); // Выводим текст
}
```

```

void loop() {
  lcd.createChar(1, simvol);           // Создаем символ под номером 1
  sensors.requestTemperatures();       // Считываем показания температуры
  lcd.setCursor(7,0);                  // Устанавливаем курсор на 1 строку, ячейка 7
  lcd.print(sensors.getTempCByIndex(0)); // Выводим показания температуры
  lcd.setCursor(12,0);                 // Устанавливаем курсор на 1 строку, ячейка 12
  lcd.print("\1");                     // Выводим символ градуса
  lcd.setCursor(13,0);                 // Устанавливаем курсор на 1 строку, ячейка 13
  lcd.print("C");                      // Выводим текст

  int DT = digitalRead(6);              // проверяем на ноль синусоиду

  err = tzad-sensors;                  // вычисляем разницу между реальной и заданной
  температурой
  if ((err!=0)&(DT==HIGH))              // задаем чтобы воздействие пошло только
                                        // при разнице температур и нуле синусоиды

  // П
  up = kp * err;
  // И
  iSum = iSum * k_down_i + err;
  ui = ki * iSum;
  // Д
  //unsigned long T = time-pastTime;
  time = millis(); // текущее время
  ud = kd * (err - oldErr)/(time-pastTime); // de/dt
  oldErr = err;
  pastTime = time;
  float u = up + ui + ud; // управляющее воздействие
  u = constrain(u, 0, 100);
  u=analogWrite(A1);

```