

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет            компьютерных систем и сетей  
Кафедра            информатики  
Дисциплина        архитектура вычислительных систем

*К защите допустить:*  
Заведующий кафедрой информатики  
\_\_\_\_\_ С.И. Сиротко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
курсовой работы  
на тему

**ТЕРМОСТАТ ДЛЯ СЕПАРАТОРА ДИСПЛЕЙНЫХ МОДУЛЕЙ НА  
БАЗЕ ARDUINO**

БГУИР КП 1-40 04 01 007 ПЗ

Студент	Б.Я. Дмитрук
Руководитель	А.А. Калиновская
Нормоконтролер	А.А. Калиновская

Минск 2024

## СОДЕРЖАНИЕ

Введение.....	5
1 Архитектура вычислительной системы.....	7
1.1 Общий обзор платформы <i>Arduino</i> .....	7
1.2 История версии.....	9
1.3 Достоинства <i>Arduino</i> .....	17
1.4 Анализ выбранной вычислительной системы для написания программы.....	18
2 Платформа программного обеспечения.....	20
2.1 Структура и архитектура платформы <i>Arduino IDE</i> .....	20
2.2 Достоинства C/C++ как языка написания программы .....	23
3 Теоретическое обоснование разработки программного продукта.....	26
3.1 Обоснование необходимости разработки.....	25
3.2 Технологии программирования, используемые для реализации поставленных задач .....	25
4 Проектирование функциональных возможностей системы .....	27
4.1 Обзор используемых компонентов системы.....	27
4.2 Функции и устройство разрабатываемого устройства.....	30
5 Архитектура разрабатываемой программы.....	33
5.1 Общая структура программы.....	33
5.2 Описание функциональной схемы алгоритма .....	33
5.3 Описание блок-схемы алгоритма .....	34
5.4 Описание интерфейса разрабатываемого устройства.....	34
5.5 Обработка ошибок .....	36
Заключение .....	38
Список литературных источников .....	39
Приложение А (обязательное) справка о проверке на заимствования .....	41
Приложение Б (обязательное) Листинг программного кода .....	42
Приложение В (обязательное) Функциональная схема алгоритма, реализующего программное средство.....	46
Приложение Г (обязательное) Блок схема алгоритма, реализующего программное средство.....	47
Приложение Д (обязательное) Графический интерфейс пользователя.....	48
Приложение Е (обязательное) Ведомость курсового проекта .....	50

## ВВЕДЕНИЕ

Термостаты представляют собой важные устройства, используемые для автоматического контроля температуры в различных системах, таких как климатические установки, отопление и вентиляция. С развитием технологий и увеличением потребностей в автоматизации, термостаты становятся неотъемлемой частью современных систем управления климатом. Они позволяют поддерживать оптимальные условия для человека и оборудования, а также способствуют экономии энергии, что актуально в условиях растущих цен на ресурсы и необходимости заботы об экологии. Конкретно в этой работе будет рассмотрен процесс создания термостата для сепаратора дисплейных модулей.

Современные термостаты могут быть реализованы на основе различных технологий, включая механические, электрические и электронные решения. Каждое из этих направлений имеет свои особенности, преимущества и недостатки. Однако, с учетом широких возможностей микроконтроллеров, таких как *Arduino*, разработка термостата на их основе становится все более популярной. Использование *Arduino* позволяет интегрировать в устройство различные сенсоры и исполнительные механизмы, а также реализовать сложные алгоритмы управления, что значительно расширяет функциональность термостата.

В рамках данного курсового проекта рассматривается разработка и реализация термостата на базе *Arduino UNO* с использованием *PTC* нагревателя и датчиков температуры. В процессе работы будет проведен анализ существующих решений в области термостатов, рассмотрены методы их работы и подключения, а также изучены основные аспекты программирования на платформе *Arduino* для управления нагревательными элементами.

Разработка и создание термостата на базе *Arduino* представляет собой проект, который требует знаний в области электроники, программирования и управления системами. Этот проект позволяет применить полученные знания на практике и открыть новые возможности в области робототехники и автоматизации. Аппаратная часть проекта представляет собой плату *Arduino UNO* в сочетании с *PTC* нагревателем, датчиками температуры, блоком питания, релейным модулем и другими электронными компонентами.

Перечень задач проекта:

1. Изучение теоретических основ работы термостатов, включая принципы терморегуляции, методы контроля температуры и физические процессы, связанные с работой датчиков измерения температуры.

2. Анализ существующих решений в области термостатов и их применения, исследование и выбор наиболее подходящих компонентов для реализации термостата на базе *Arduino*.

3. Разработка аппаратной части термостата, включая выбор и подключение необходимых компонентов, таких как реле, датчики температуры и нагревательные элементы, а также плата *Arduino* для управления системой.

4. Программирование *Arduino* для реализации алгоритмов управления термостатом, включая разработку и оптимизацию алгоритмов обработки данных от датчиков и управления нагревателем.

5. Тестирование и отладка системы термостата, проверка его способности поддерживать заданные температурные параметры в различных условиях.

6. Анализ полученных результатов и оценка стабильности и эффективности созданного термостата на базе *Arduino*.

Выполнение этих задач позволит достичь поставленной цели – создать эффективный термостат на базе *Arduino* и исследовать его работу, а также возможности для дальнейшего развития и усовершенствования системы.

Данная пояснительная записка к курсовому проекту выполнена по стандартам предприятия БГУИР [1].

# 1 АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

## 1.1 Общий обзор платформы *Arduino*

*Arduino* представляет собой мощную платформу для проектирования и разработки электронных устройств, обеспечивающую взаимодействие с физической средой через простые в использовании средства программирования и аппаратного обеспечения. Эта открытая платформа была разработана для облегчения процесса создания интерактивных объектов и устройств, позволяя разработчикам и любителям заниматься "физическим вычислением" (*physiCal Computing*). Важной особенностью *Arduino* является ее доступность, что делает ее идеальным инструментом как для начинающих, так и для опытных инженеров.

Существует несколько моделей плат *Arduino*, каждая из которых обладает уникальными характеристиками и функциональными возможностями. Наиболее известные из них включают *Arduino Uno*, *Arduino Mega*, *Arduino Nano* и *Arduino Leonardo*.

Например, *Arduino Uno* является одной из самых популярных плат и основана на микроконтроллере *ATmega328P*, который имеет 32 КБ встроенной памяти и поддерживает до 16 МГц тактовую частоту [2]. Эта плата оснащена 14 цифровыми входами/выходами (из которых 6 могут использоваться в качестве ШИМ-выходов) и 6 аналоговых входами, что делает ее подходящей для большинства базовых проектов.

*Arduino Mega* представляет собой более мощную версию, предназначенную для более сложных приложений. Она основана на микроконтроллере *ATmega2560*, который предлагает 256 КБ памяти, 54 цифровых входа/выхода и 16 аналоговых входов. Благодаря этому, *Arduino Mega* идеально подходит для проектов, требующих большого количества подключенных устройств и сложной логики.

*Arduino Nano*, в свою очередь, представляет собой компактную версию, которая сохраняет основные характеристики *Arduino Uno*, но в уменьшенном формате. Она также основана на *ATmega328P* и идеально подходит для проектов, где пространство ограничено, например, в носимых устройствах или в компактных автоматизированных системах [3].

Модель *Arduino Leonardo* отличается от других плат тем, что она может эмулировать *USB*-устройства, такие как клавиатуры или мыши, что открывает дополнительные возможности для взаимодействия с пользователем и другими устройствами. Это достигается за счет использования микроконтроллера *ATmega32u4*, который имеет встроенный *USB*-интерфейс.

Платформа *Arduino* поддерживает широкий спектр датчиков и исполнительных механизмов, что позволяет реализовывать множество проектов – от простых световых сигналов до сложных систем управления. Датчики температуры, такие как *DHT11* и *DHT22*, используются для измерения температуры и влажности, в то время как датчики движения, такие как *HC-SR501*, позволяют обнаруживать движения в пределах заданной зоны.

Исполнительные механизмы, такие как серводвигатели и шаговые двигатели, могут быть легко интегрированы в проекты *Arduino*. Например, сервоприводы *SG90* широко используются в робототехнике для создания механизмов управления положением.

С точки зрения программного обеспечения, *Arduino* использует среду разработки *Arduino IDE*, которая предоставляет пользователям удобный интерфейс для написания и загрузки кода на платы. Язык программирования основан на C/C++, что делает его понятным для разработчиков, знакомых с этими языками. Кроме того, доступ к обширной библиотеке готовых библиотек и примеров кода позволяет пользователям быстро осваивать новые концепции и технологии.

Концепция *Arduino* не предусматривает наличия встроенного корпуса или стандартных методов монтажа. Пользователю предоставляется свобода в выборе способов установки и механической защиты плат, будь то самостоятельная реализация или использование сторонних решений

Помимо этого, независимые производители выпускают разнообразные датчики и исполнительные устройства, которые в значительной степени совместимы с *Arduino*. Стандартные платы *Arduino* и их совместимые аналоги спроектированы для модульного наращивания посредством штыревых разъемов, что позволяет пользователю добавлять необходимые периферийные устройства и внешние подключения к базовой микропроцессорной плате (рисунок 1.1).

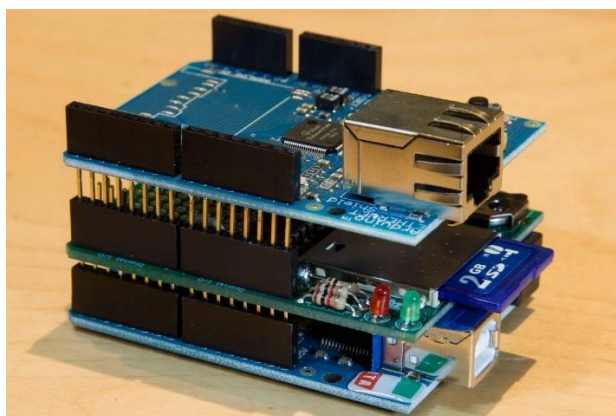


Рисунок 1.1 – Классический конструктив *Arduino*

В силу всех перечисленных выше преимуществ *Arduino* представляет собой универсальный инструмент для создания электронных устройств и систем, позволяя разработчикам реализовывать инновационные решения в самых различных областях, от автоматизации до робототехники. Благодаря доступности и простоте использования, *Arduino* продолжает набирать популярность как среди образовательных учреждений, так и среди любителей электроники по всему миру.

## 1.2 История версии

В настоящее время существует множество значимых разработок на базе платформы *Arduino*, включая такие проекты, как алкотестеры, светодиодные кубы, системы автоматизации домашних процессов, дисплеи для отображения сообщений из *Twitter*, а также наборы для анализа ДНК. *Arduino* нашла широкое применение как в любительской, так и в профессиональной среде, благодаря чему образовались целые сообщества и клубы энтузиастов, посвященные изучению и развитию проектов на базе данной платформы. Например, компания Google разработала комплект разработчика для устройств на базе *Android*, который основан на *Arduino*. По словам Дейла Доэрти (*Dale Dougherty*), редактора и издателя журнала "*Make*", который считается ведущим изданием для любителей *DIY*-проектов, платформа *Arduino* стала своего рода "мозговым центром" для создателей и инженеров.

В 2008 году Массимо Банци совместно с партнёрами основал в Соединённых Штатах компанию *Arduino LLC*, которая на сегодняшний день известна как среди любителей электроники, так и среди профессионалов в области прототипирования. Проект *Arduino* представляет собой электронную платформу, отличающуюся простотой аппаратной части, интуитивно понятным языком программирования и использованием открытого исходного кода.

Изначально платы *Arduino* разрабатывались для начинающих инженеров-схемотехников, заинтересованных в создании собственных электронных устройств, каждое из которых могло бы служить основой для инновационных стартапов. На данный момент компания *Arduino* и её конкуренты предлагают широкий ассортимент плат, начиная от простых микроконтроллеров для базовых прототипов и заканчивая сложными устройствами, способными управлять мощными электродвигателями.

Проект *Arduino* был воспринят с энтузиазмом профессиональным сообществом разработчиков по всему миру. На базе этих микроконтроллеров с простым языком программирования были созданы многочисленные

прототипы различных электронных устройств. Долгое время производство плат *Arduino* осуществлялось исключительно на заводе компании *Smart Projects SRL* в Италии, что позволило продукции выделяться высоким качеством и надёжностью. Массимо Банци и его команда активно совершенствовали как аппаратную, так и программную составляющую платформы, а итальянская сторона занималась производством оборудования, передавая процент от продаж создателям бренда [4].

Однако с ростом популярности платформы *Arduino* возникла конкуренция, и в США было принято решение, что производство плат должно быть расширено за пределы Италии, в том числе в страны Юго-Восточной Азии. Это решение способствовало снижению себестоимости продукции и расширению её доступности.

Документация, прошивочное обеспечение и схемы плат *Arduino* распространяются под лицензией *Creative Commons Attribution ShareAlike 3.0*, обеспечивая свободный доступ через официальный сайт проекта. Чертежи печатных плат для некоторых версий *Arduino* также открыты для пользователей. Исходный код интегрированной среды разработки (*IDE*) распространяется под лицензией *GPLv2*, в то время как используемые библиотеки лицензируются в соответствии с *LGPL*.

Несмотря на то, что аппаратная документация и программное обеспечение доступны по лицензии «*Copyleft*», разработчики установили ограничения на использование торговой марки «*Arduino*» и её производных. Согласно официальному документу по использованию бренда *Arduino*, проект остаётся открытым для всех, кто заинтересован в работе над официальными продуктами, однако использование имени «*Arduino*» без разрешения для производных проектов не допускается.

Платформа *Arduino* имеет несколько версий. Одна из последних модификаций, *Arduino Leonardo*, построена на базе микроконтроллера *ATMega32u4*. Версии *Uno* и предшествующая ей *Duemilanove* базируются на микроконтроллере *Atmel ATMega328*. Более ранние версии, такие как *Diecimila* и первая рабочая версия платформы, использовали микроконтроллер *Atmel ATMega168*, а самые ранние – *ATMega8*. Вариант *Arduino Mega2560* основан на микроконтроллере *ATMega2560*, обеспечивая значительно более высокие технические возможности для сложных проектов.

Рассмотрим следующие модели подробнее:

*Arduino Nano* представляет собой компактную отладочную плату, которая занимает одну из ведущих позиций по популярности среди радиолюбителей и программистов. Несмотря на свои небольшие размеры, данное устройство практически не уступает известной *Arduino Uno* по



функциональным возможностям и может быть эффективно применено в проектах, где размеры компонентов имеют критическое значение [5].

Ранние версии *Arduino Nano* были основаны на микроконтроллере ATmega168. Начиная с версии 3.0, в конструкции платы используются более современные микроконтроллеры *ATmega328*, обладающие увеличенным объемом *FLASH* и *EEPROM*-памяти, а также обеспечивающие более высокую тактовую частоту. Внешний вид платы *Arduino Nano V3.0* представлен на рисунке 1.2.

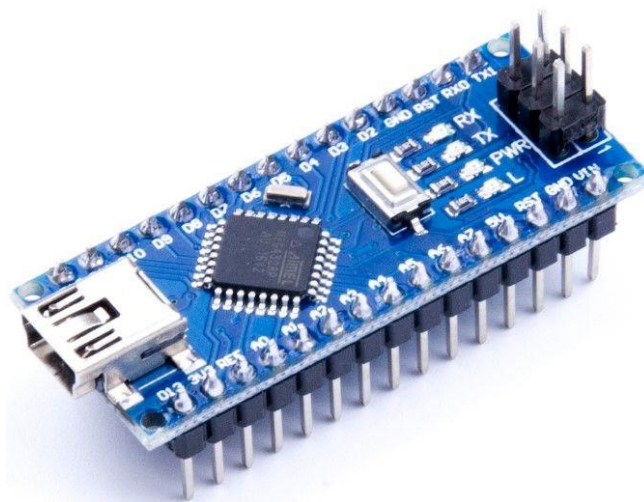


Рисунок 1.2 – Плата *Arduino Nano*

На основании представленного выше рисунка можно отметить, что для взаимодействия с внешними устройствами на плате предусмотрены штыревые колодки. Их наличие значительно упрощает процесс макетирования, однако, при необходимости, их можно исключить из сборки, выполнив прямую пайку проводов к соответствующим контактам. Эти колодки также играют важную роль при использовании специализированных плат расширения (шилдов), разработанных в большом количестве для данной версии *Arduino*.

При детальном рассмотрении платы *Arduino Nano* можно обнаружить не только основной микроконтроллер *ATmega328*, но и множество вспомогательных компонентов, обеспечивающих стабильную работу всей системы. На рисунке 1.3 представлены компоненты платы с указанием их функционального назначения.

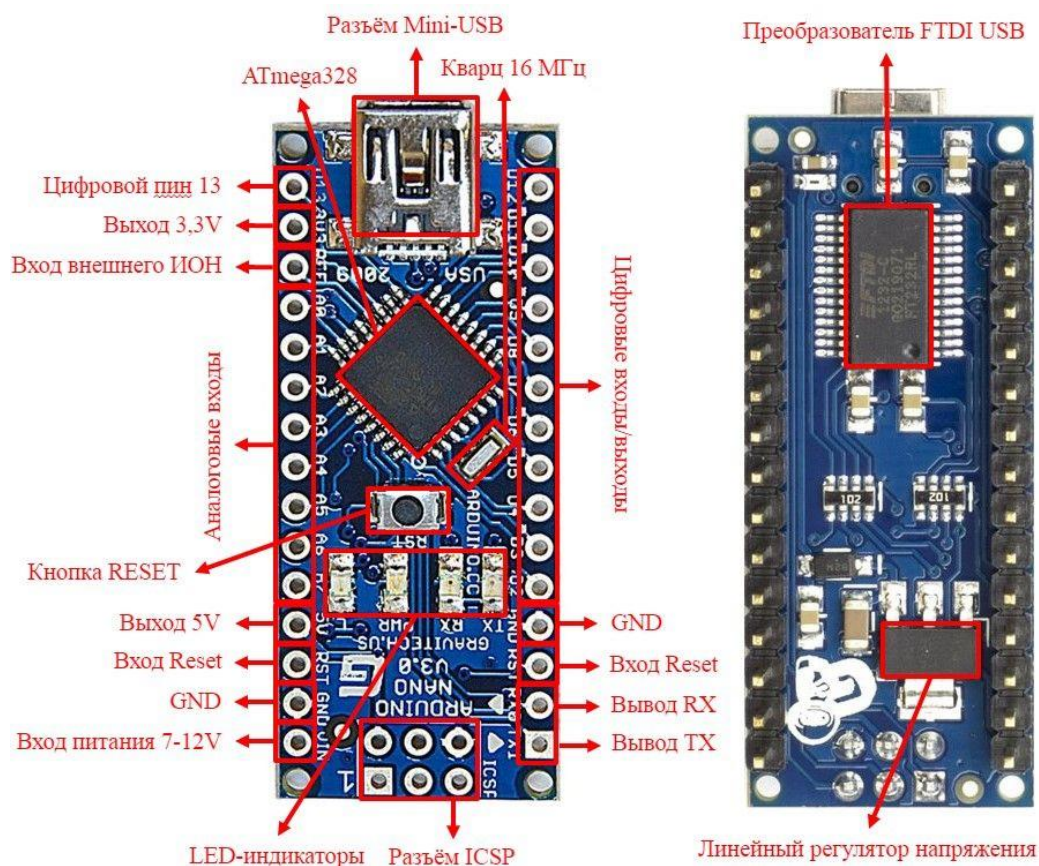


Рисунок 1.3 – Компоненты *Arduino Nano*

Для оптимизации пространства на плате разработчики разместили радиоэлементы с обеих сторон. На лицевой стороне располагаются все основные компоненты и элементы шелкографии, включая микроконтроллер *ATmega328*, кварцевый резонатор, разъём *Mini-USB*, кнопку сброса, а также четыре индикаторных светодиода: *TX*, *RX*, *PWR* и *L*. Светодиоды *TX* и *RX* сигнализируют о передаче и приёме данных через последовательный порт, светодиод *PWR* указывает на наличие питания на плате, а индикатор *L*, связанный с выводом №13, используется для отображения состояния высокого уровня сигнала.

На обратной стороне платы размещены такие важные компоненты, как линейный стабилизатор напряжения на 5V и преобразователь интерфейса FTDI USB, обеспечивающие стабильность работы устройства и поддержку взаимодействия с компьютером.

*Arduino Uno* представляет собой отладочный комплекс на основе микроконтроллера *ATMega328*. Эта плата выполняет функцию интерфейса между пользователем и микроконтроллером, обеспечивая удобный доступ к его выводам и возможность загрузки прошивки непосредственно из среды программирования. Помимо основной функциональности, плата обладает рядом дополнительных возможностей. Благодаря продуманной конструкции,

компактным размерам, а также обширной библиотеке примеров и кода, Arduino Uno стала популярной среди разработчиков электронных устройств. В настоящее время в интернете можно найти множество проектов, использующих эту плату в качестве основы.

Разработанная итальянскими инженерами, Arduino Uno позиционируется как одна из ключевых плат с открытой архитектурой. В результате её популярности на рынке появилось множество клонов, полностью совместимых как по программному, так и по аппаратному обеспечению. На рисунке 1.4 представлен внешний вид платы Arduino Uno с обеих сторон, демонстрирующий её конструктивные особенности.

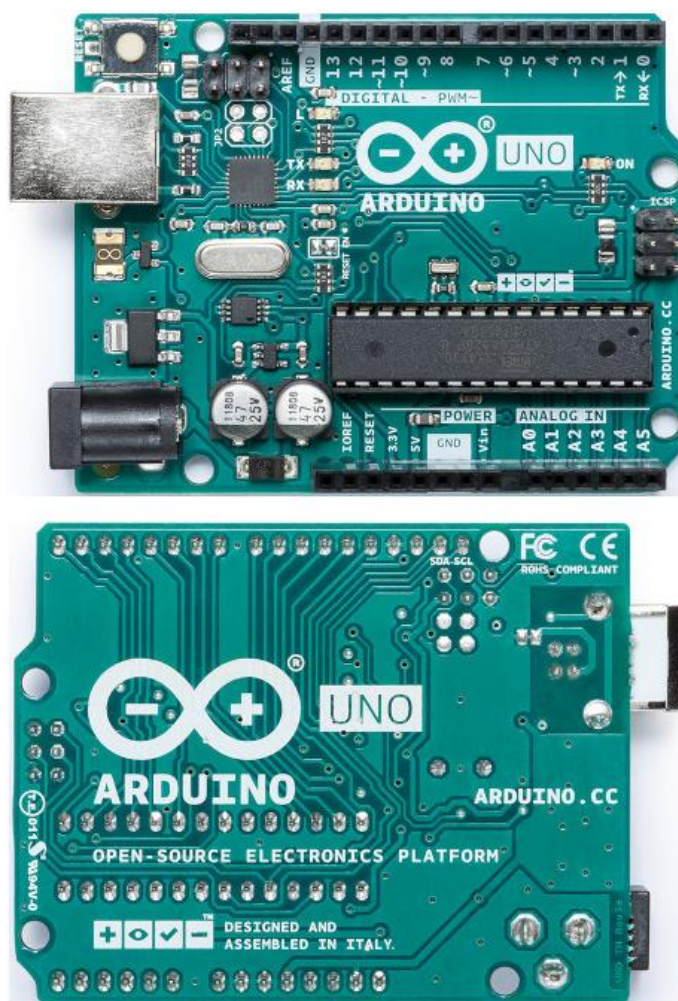


Рисунок 1.4 – Плата Arduino Uno

На основании представленного рисунка можно отметить, что подключение к выводам микроконтроллера осуществляется через штыревые разъёмы, расположенные по обе стороны платы. Такое конструктивное решение позволяет разработчику интегрировать *ATMega328* с внешними устройствами с использованием макетных проводов. Кроме того, под



архитектуру *Arduino Uno* разработано значительное количество шилдов, обеспечивающих расширение функциональных возможностей устройства посредством каскадного подключения. Пример каскадного включения шилдов продемонстрирован на рисунке 1.5.

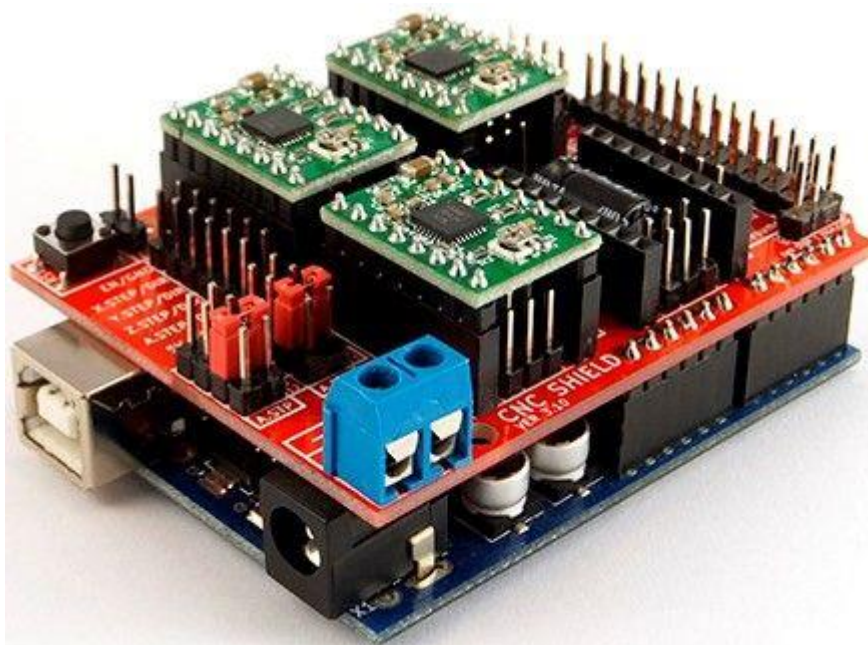


Рисунок 1.5 – *Arduino Uno* и шилд для ЧПУ

Подобный подход значительно упрощает и ускоряет процесс создания прототипов различных электронных устройств, превращая трудоёмкие этапы разработки в удобный процесс сборки конструктора. В настоящее время существует широкий ассортимент шилдов, таких как шилды с датчиками, клавиатуры, дисплеи, расширители портов, радиомодули и многие другие устройства, способные удовлетворить потребности даже самых требовательных разработчиков.

Однако доступ к портам микроконтроллера – лишь одна из особенностей *Arduino Uno*. Детальная архитектура компонентов представлена на рисунке 1.6, где все ключевые элементы платы выделены и снабжены пояснениями, включая детализированную характеристику штыревых разъёмов.

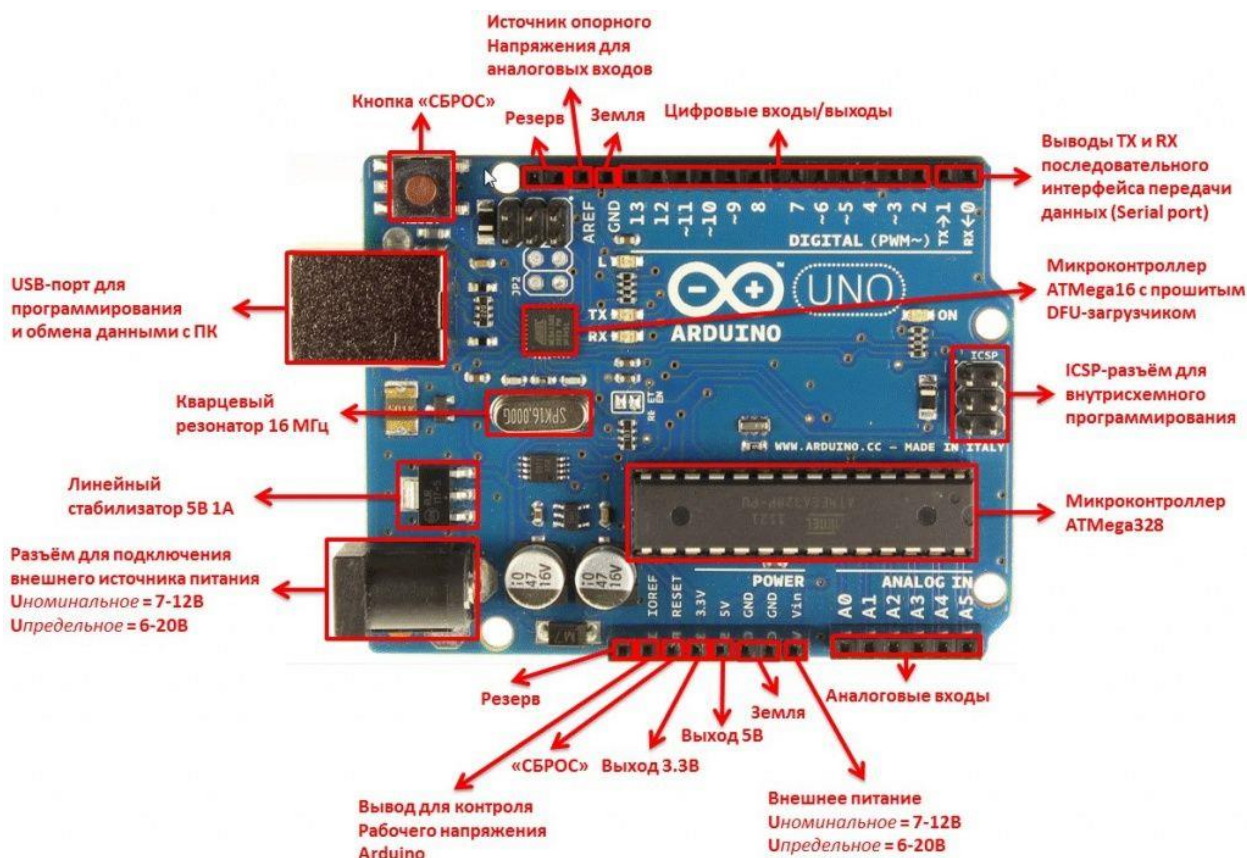


Рисунок 1.6 – Компоненты *Arduino Uno*

*Arduino Mega 2560 R3* является одной из наиболее мощных и универсальных платформ в семействе Arduino, предназначенной для реализации сложных проектов в области электроники и автоматизации. Основанная на микроконтроллере *ATmega2560*, эта плата предоставляет расширенные возможности по сравнению с другими моделями, что делает её особенно подходящей для задач, требующих большого количества входов и выходов, а также значительных вычислительных ресурсов [6].

Плата работает на напряжении 5 В и поддерживает 54 цифровых ввода/вывода, из которых 15 могут использоваться в качестве выходов с широтно-импульсной модуляцией. Это позволяет реализовывать сложные алгоритмы управления и взаимодействия с различными устройствами. Кроме того, *Arduino Mega 2560 R3* оснащена 16 аналоговыми входами, что значительно расширяет возможности по сбору данных с различных сенсоров и их последующей обработке.



Рисунок 1.7 – Плата Arduino Mega 2560 R3

Ключевой особенностью данной модели является её память. Она включает в себя 256 КБ флэш-памяти, из которых 8 КБ резервированы для загрузчика, 8 КБ оперативной памяти *SRAM* и 4 КБ энергонезависимой памяти *EEPROM*. Такой объём памяти позволяет обрабатывать большие объёмы данных и сохранять критически важную информацию даже после отключения питания.

*Arduino Mega 2560 R3* поддерживает различные протоколы связи, включая *UART*, *I2C* и *SPI*. Это обеспечивает гибкость при интеграции с другими электронными модулями и устройствами, что особенно важно в сложных системах автоматизации и робототехники. Благодаря этому, плата может функционировать как центральный элемент в системах сбора и обработки данных, управления исполнительными механизмами и обмена информацией между различными компонентами системы.



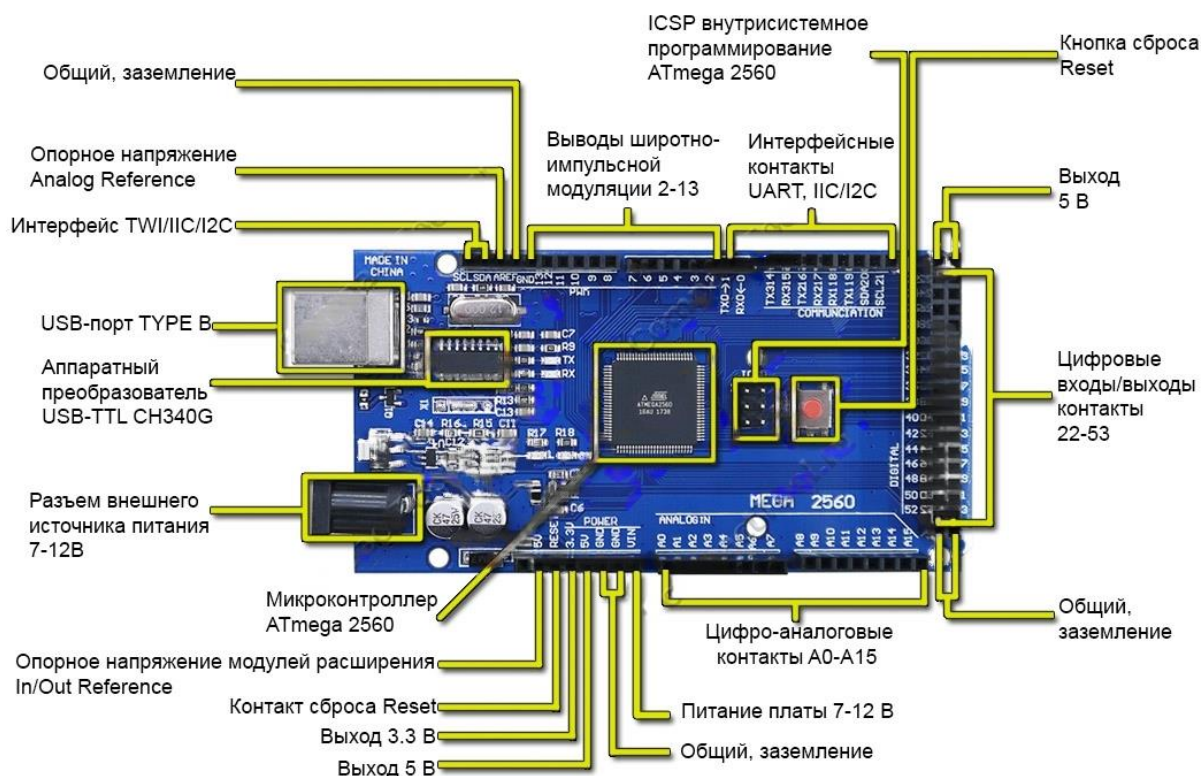


Рисунок 1.8 – Компоненты *Arduino Mega 2560 R3*

Плата совместима с широким спектром дополнительных модулей и шилдов, что позволяет без труда расширять её функциональные возможности. Такая совместимость делает её идеальной платформой для научных исследований, прототипирования и создания интерфейсных систем.

### 1.3 Достоинства *Arduino*

Платформа *Arduino* зарекомендовала себя как одна из наиболее популярных и удобных сред для разработки электронных устройств благодаря своей доступности, простоте использования и широкому сообществу пользователей. Эти достоинства делают *Arduino* особенно привлекательной для новичков в области схемотехники и разработки прототипов, а также для профессиональных инженеров, занимающихся созданием сложных систем. Рассмотрим ключевые достоинства платформы *Arduino*, а также причины выбора модели *Arduino Uno* для реализации проекта термостата:

1. Простота и доступность. *Arduino* предлагает пользователям простую и интуитивно понятную среду разработки, что делает её идеальной для новичков. Интегрированная среда разработки (*IDE*) поддерживает простые языковые конструкции, позволяя быстро начинать программирование без

необходимости изучать сложные языки программирования. Это особенно важно при разработке термостата, где требуется быстрое создание и тестирование программного обеспечения для управления температурой.

2. Открытая архитектура. Открытая архитектура *Arduino* предоставляет возможность свободного доступа к схемам, прошивкам и документации. Это позволяет разработчикам не только модифицировать существующие решения, но и создавать собственные уникальные проекты. В контексте термостата это означает, что можно легко адаптировать уже существующие библиотеки и примеры, что значительно ускоряет процесс разработки.

3. Широкий ассортимент библиотек и примеров. Платформа *Arduino* располагает обширной библиотекой готовых решений и примеров кода, что упрощает процесс создания прототипов. Для разработки термостата доступны библиотеки, которые позволяют работать с различными датчиками температуры, например, с датчиками *LM35* или *DHT22*, а также модулями управления реле, что необходимо для управления нагревателями. Это экономит время на написание кода с нуля и помогает сосредоточиться на реализации уникальных функций термостата.

4. Широкое сообщество и поддержка. Одним из главных достоинств *Arduino* является его активное сообщество пользователей и разработчиков. Форумы, блоги и онлайн-ресурсы содержат множество полезной информации, включая советы по решению проблем и рекомендации по улучшению проектов. При разработке термостата, если возникнут сложности, разработчик всегда сможет найти помощь и поддержку в сообществе *Arduino*.

5. Модульность и совместимость. Платформа *Arduino* предоставляет множество модулей и шилдов, которые можно использовать для расширения функционала проекта. В случае термостата можно интегрировать дополнительные датчики, например, для контроля влажности, или модули связи, такие как *Wi-Fi* или *Bluetooth*, для удалённого управления. Модульность позволяет адаптировать термостат под конкретные требования пользователя.

6. Низкая стоимость. Одним из значительных факторов, способствующих популярности платформы *Arduino*, является её доступная цена. Модель *Arduino Uno* является одной из самых бюджетных и одновременно мощных плат, что делает её идеальным выбором для разработки прототипов, особенно для учебных проектов и стартапов. Это позволяет снизить общие затраты на разработку термостата, не жертвуя при этом качеством.

#### **1.4 Анализ выбранной вычислительной системы для написания программы**



Среди всех моделей *Arduino*, выбор *Arduino Uno* в контексте разработки термостата обусловлен рядом факторов, которые подчеркивают её оптимальное сочетание возможностей и функциональности.

1. Достаточная мощность и функциональность. *Arduino Uno* основана на микроконтроллере *ATmega328*, который обладает достаточной вычислительной мощностью для выполнения необходимых алгоритмов управления температурой. Для термостата требуется обработка данных с датчиков температуры и управление нагревателями, что вполне соответствует возможностям данного микроконтроллера.

2. Простота подключения и использования. *Arduino Uno* имеет четко организованные пины, что облегчает подключение внешних компонентов, таких как датчики и реле. Благодаря этому разработка термостата становится более простой и удобной, так как нет необходимости в сложных схемах подключения. Штыревые разъёмы обеспечивают надежное соединение и упрощают работу с макетной платой.

3. Поддержка аналоговых входов. Модель *Arduino Uno* обладает несколькими аналоговыми входами, что позволяет напрямую подключать аналоговые датчики температуры, такие как *LM35*. Это упрощает процесс считывания данных и их последующей обработки, делая разработку термостата более эффективной.

4. Доступность библиотек и документации. Для *Arduino Uno* доступно множество библиотек, которые упрощают взаимодействие с различными датчиками и модулями. Это позволяет разработчику сосредоточиться на логике управления, а не на низкоуровневой реализации. Документация и примеры кода для *Uno* легко доступны, что значительно ускоряет процесс разработки.

Благодаря своим многочисленным достоинствам, становится идеальным выбором для разработки термостата. Простота использования, открытая архитектура, наличие библиотек и активное сообщество способствуют успешной реализации проектов. Выбор модели *Arduino Uno* для создания термостата обусловлен её достаточной мощностью, удобством подключения и расширяемостью.

## 2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 2.1 Структура и архитектура платформы *Arduino IDE*

*Arduino Integrated Development Environment (IDE)* представляет собой специализированную программную среду для разработки, основанную на языке программирования C++. Она предназначена для создания программного обеспечения для всех плат, входящих в экосистему *Arduino*. Аббревиатура "*IDE*" расшифровывается как *Integrated Development Environment*, что в переводе на русский язык означает "интегрированная среда разработки". Использование данной среды значительно упрощает и ускоряет процесс написания программ, что является важным аспектом в разработке проектов, основанных на *Arduino*. Иллюстрация графического интерфейса *Arduino IDE* представлена на рисунке 2.1.

Одним из ключевых преимуществ *Arduino IDE* является возможность написания кода в удобном текстовом редакторе, который обеспечивает функционал, способствующий быстрому созданию программ. В отличие от традиционных текстовых редакторов, *Arduino IDE* предлагает специализированные инструменты, такие как синтаксическая подсветка, автоматическое дополнение кода и встроенные функции для компиляции и загрузки программ на целевые устройства. Это позволяет разработчикам не только повысить продуктивность, но и минимизировать количество ошибок, связанных с ручным редактированием кода.

*Arduino IDE* позволяет пользователям компилировать написанные программы в машинный код, который затем может быть загружен на любые версии плат *Arduino*, включая *Arduino UNO*, *Arduino Nano*, и многие другие (рисунок 2.1). Интерфейс данной среды отличается относительной простотой, что делает его доступным для пользователей с различным уровнем подготовки. Основой программирования в *Arduino IDE* является упрощённая версия языка C++, который сохраняет многие концепции и синтаксические структуры своего более сложного предшественника [7].

Как и в других языках, основанных на C, в *Arduino* существует ряд правил написания кода, которые необходимо соблюдать. Язык программирования, используемый в *Arduino IDE*, также является строго типизированным и компилируемым, что позволяет предотвращать многие ошибки на этапе разработки.

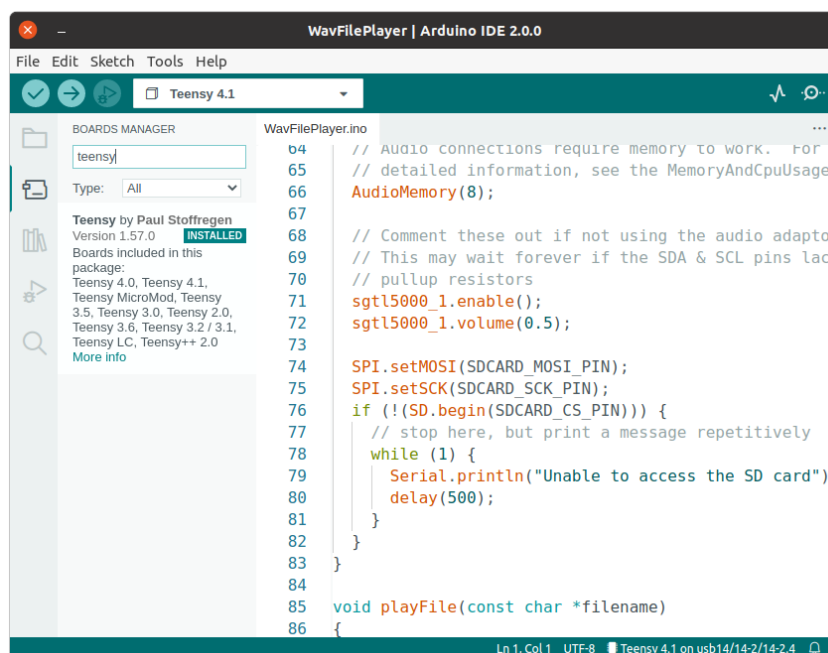


Рисунок 2.1 – Графический интерфейс *Arduino IDE*

Основные ключевые слои, обеспечивающие взаимодействие между пользователем, средой и микроконтроллером:

**2.1.1 Редактор кода.** Редактор кода является основным интерфейсом для создания и редактирования скетчей. Он поддерживает синтаксис языков программирования *C* и *C++*, адаптированный для микроконтроллеров *Arduino* с использованием специализированных библиотек и функций. В редакторе реализованы базовые функции, такие как подсветка синтаксиса, автодополнение, поиск по коду и другие инструменты, способствующие упрощению процесса написания программного кода. Код, создаваемый в редакторе, сохраняется в файлах с расширением *.ino*. Каждый проект в среде *Arduino IDE* организован в виде директории, содержащей один или несколько файлов указанного типа.

**2.1.2 Система компиляции.** *Arduino Core* представляет собой набор библиотек, обеспечивающих доступ к функциональным возможностям микроконтроллеров и позволяющих взаимодействовать с аппаратными компонентами плат (такими как выходы ввода-вывода, таймеры и прерывания). В процессе написания кода в редакторе среда разработки автоматически компилирует его, связывая с соответствующими библиотеками *Arduino Core* в зависимости от выбранной платы (например, *Arduino Nano*, *Uno*, *Mega*). По завершении компиляции генерируется машинный код,

который может быть загружен на микроконтроллер с помощью встроенного загрузчика (*bootloader*).

**2.1.3** Инструменты загрузки кода на микроконтроллер. Загрузчик (*Bootloader*) представляет собой небольшую программу, находящуюся в памяти микроконтроллера, которая управляет процессом загрузки пользовательского кода через интерфейс *USB*. *Bootloader Arduino* упрощает программирование платы, исключая необходимость подключения внешнего программатора. 3.2. Среда *Arduino IDE* использует библиотеку *avrdude* для загрузки скомпилированного кода на микроконтроллер через *USB*-интерфейс [8].

**2.1.4** Менеджер библиотек. *Arduino IDE* включает в себя менеджер библиотек, который позволяет пользователю легко подключать и устанавливать дополнительные библиотеки для работы с разнообразными датчиками, модулями и периферийными устройствами. Это упрощает процесс разработки и дает возможность быстро добавлять новый функционал в проекты. 4.2. В частности, доступны библиотеки для работы с сетевыми интерфейсами (*Wi-Fi*, *Ethernet*), дисплеями, двигателями, *GPS*-модулями, а также датчиками температуры и влажности.

**2.1.5** Монитор последовательного порта. Данная утилита встроена в среду *Arduino IDE* и предоставляет возможность пользователю получать информацию от микроконтроллера через последовательный интерфейс (*Serial*). Данные могут отображаться в виде текста, чисел или других форматов, что значительно облегчает процесс отладки и мониторинга работы устройства в реальном времени [9]. Пользователь может отправлять команды или данные на микроконтроллер через последовательный интерфейс, что позволяет организовать двустороннюю связь для управления устройством.

**2.1.6** Менеджер плат. *Arduino IDE* поддерживает разнообразные платы *Arduino* и совместимые с ними микроконтроллеры. Для удобства работы с различными аппаратными платформами в среде интегрирован менеджер плат, который упрощает выбор и подключение плат, а также загрузку соответствующих драйверов и библиотек. 6.2. Менеджер плат позволяет настраивать параметры компиляции и загрузки кода в зависимости от выбранной аппаратной платформы. Например, для *Arduino Nano* используется микроконтроллер *ATmega328*, в то время как для плат типа *Arduino Due* применяются более мощные микроконтроллеры на базе *ARM*.

Таким образом *Arduino IDE* представляет собой мощный инструмент для разработчиков, предлагая функционал, который значительно упрощает процесс создания программного обеспечения для плат *Arduino*. Благодаря своей доступности, простоте в использовании и широкому спектру поддерживаемых устройств, эта среда стала неотъемлемой частью сообщества *Arduino* и важным элементом в разработке электронных проектов.

## **2.2 Достоинства C/C++ как языка написания программы**

Языки программирования C и C++ зарекомендовали себя как одни из наиболее подходящих инструментов для разработки программного обеспечения для встроенных систем, в частности, в контексте платформы *Arduino*. Их применение в проекте по разработке термостата имеет ряд весомых причин и достоинств, которые будут подробно рассмотрены ниже.

**2.2.1 Производительность и эффективность.** C и C++ обеспечивают высокий уровень производительности, что является критически важным для встраиваемых систем, таких как термостаты. Эти языки позволяют разработчикам непосредственно управлять ресурсами микроконтроллера, такими как память и процессорное время, что в свою очередь способствует оптимизации работы приложений. Данный аспект особенно важен при реализации алгоритмов управления температурой, которые должны быть высокоэффективными и быстрыми для своевременной реакции на изменения условий.

**2.2.2 Близость к аппаратному обеспечению.** Языки C и C++ предоставляют возможность низкоуровневого программирования, что позволяет разработчикам взаимодействовать с аппаратными компонентами напрямую. Это включает управление портами ввода-вывода, настройку регистров и работу с периферийными устройствами. В проекте термостата это позволяет, например, эффективно управлять датчиками температуры и исполнительными механизмами, такими как реле и нагревательные элементы, что критически важно для точности и надежности работы устройства.

**2.2.3 Широкий спектр библиотек и инструментов.** Для языка C/C++ существует обширный набор библиотек, специально разработанных для работы с платформой *Arduino*. Эти библиотеки содержат готовые функции и методы, которые упрощают разработку, позволяя сосредоточиться на логике приложения, а не на реализации низкоуровневых операций. Например, использование библиотек для работы с датчиками температуры, управления

ЖК-дисплеями или интерфейсами связи существенно ускоряет процесс разработки и снижает вероятность ошибок.

**2.2.4** Портативность кода. *C* и *C++* являются языками с высокой степенью портативности. Код, написанный на этих языках для одной платформы, как правило, может быть легко адаптирован для работы на других системах и микроконтроллерах. Это свойство делает проекты, разработанные на *C/C++*, более устойчивыми к изменениям аппаратной платформы, что особенно важно в условиях быстро меняющейся технологии и возможных обновлений в будущем.

**2.2.5** Поддержка сообществом и документация. Языки *C* и *C++* обладают широким сообществом разработчиков и обширной документацией, что позволяет находить решения для возникающих проблем и получать поддержку. В контексте разработки термостата, наличие активного сообщества может быть неопределимо для обмена опытом, получения рекомендаций и изучения лучших практик.

**2.2.6** Устойчивость и надежность. Языки *C* и *C++* используются в различных критически важных приложениях, включая системы управления, медицинское оборудование и автомобили. Это говорит о высокой степени надежности и устойчивости, что делает их подходящими для разработки термостатов, где требуется гарантированная работа и безопасность.

Таким образом, использование языков *C* и *C++* для разработки проекта термостата на платформе *Arduino UNO* предоставляет разработчикам мощные инструменты для создания эффективного, надежного и портируемого программного обеспечения. Благодаря своим достоинствам, эти языки становятся неотъемлемой частью процессов проектирования и реализации встроенных систем, обеспечивая достижение поставленных целей в области управления температурой и других связанных задач.

## **3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА**

### **3.1 Обоснование необходимости разработки**

Разработка термостата для сепаратора дисплеев представляет собой актуальную задачу в условиях современного рынка ремонта мобильных устройств. Сепарация стекла от матрицы является важным этапом при восстановлении функциональности экранов, и от качества данного процесса зависит дальнейшая работоспособность устройства. Традиционные методы отклеивания стекол зачастую сопряжены с рисками повреждения как матрицы, так и самого стеклянного покрытия. Это обуславливает необходимость внедрения инновационных решений, направленных на оптимизацию процесса сепарации.

Термостат, предназначенный для регулирования температуры в процессе отклеивания, способен значительно повысить эффективность и безопасность процедуры. С точки зрения физики, применение тепла способствует снижению адгезионных свойств клеевых составов, используемых для фиксации стекла. Однако контроль за температурным режимом в процессе обработки является критически важным, так как перегрев может привести к повреждению электронных компонентов, а недостаточный нагрев не обеспечит необходимого эффекта. Таким образом, термостат играет ключевую роль в достижении оптимальных условий для сепарации.

Кроме того, автоматизация процесса с использованием термостата способствует повышению производительности и снижению временных затрат на обслуживание, что является важным фактором в условиях конкурентной борьбы на рынке сервисных услуг. Внедрение такого устройства позволит не только повысить качество выполняемых работ, но и создать предпосылки для стандартизации процесса ремонта, что, в свою очередь, может стать конкурентным преимуществом для сервисных центров.

### **3.2 Технологии программирования, используемые для реализации поставленных задач**

Для успешной разработки термостата на базе *Arduino Uno* применяются такие технологии программирования как *Arduino IDE*, языки программирования *C/C++*, библиотеки *Arduino* и специализированные алгоритмы:

**3.2.1 Arduino IDE.** *Arduino Integrated Development Environment (IDE)* представляет собой официальную среду разработки, специально созданную для программирования платформы *Arduino*. Данная среда предлагает простой и интуитивно понятный интерфейс, который позволяет загружать программный код на плату *Arduino*. *Arduino IDE* основан на языке программирования *C/C++*, а также предоставляет набор функций и библиотек, специально разработанных для работы с различными компонентами *Arduino*.

**3.2.2 Язык программирования C/C++.** Поскольку *Arduino* основан на языке программирования *C/C++*, знание и понимание его основных концепций и синтаксиса являются необходимыми для разработки термостата. Язык *C/C++* предоставляет мощные инструменты для низкоуровневого программирования, такие как управление портами ввода-вывода, обработка прерываний и работа с памятью, что особенно актуально при взаимодействии с датчиками и нагревательными элементами.

**3.2.3 Библиотеки Arduino.** *Arduino IDE* включает в себя множество готовых библиотек, которые значительно упрощают процесс разработки проектов на *Arduino*. В контексте термостата важными библиотеками могут быть библиотеки для работы с температурными датчиками, такими как *LM35*, а также библиотеки для управления реле и другими устройствами, необходимыми для функционирования термостата.

**3.2.4 Алгоритмы управления температурой.** Для реализации функциональности термостата необходимо разработать алгоритмы, которые будут анализировать данные с температурных датчиков и управлять нагревательным элементом в соответствии с заданной температурой. Здесь могут быть использованы алгоритмы, такие как *PID* регуляторы, которые обеспечивают точное и стабильное управление температурой, минимизируя колебания.

Таким образом, для разработки термостата на базе *Arduino Uno* были использованы *Arduino IDE*, язык *C/C++*, библиотеки *Arduino* и алгоритмы управления температурой. Эти технологии обеспечили удобное программирование, эффективное управление оборудованием и стабильную работу устройства, позволяя точно контролировать температуру и гарантировать надёжность системы.





широко применяется для нагрева воздуха и твердых поверхностей благодаря своей способности автоматически регулировать выходную мощность в зависимости от температуры окружающей среды. Данный тип нагревателя эффективен в системах, требующих стабильного и безопасного нагрева, так как его свойства позволяют избежать перегрева и обеспечивают равномерное распределение тепла по всей рабочей поверхности [14]. Внешний вид элемента представлен на рисунке 4.2.

Основные технические характеристики [15]:

- Входное напряжение: 12В постоянного тока.
- Максимальная мощность: 30 Вт.
- Максимальная достигаемая температура: 120°C
- Размеры: 35 x 20 x 5 мм.
- Нагревательный материал: *PTC* термистор.
- Способ передачи тепла: теплопроводность.
- Изоляция поверхности: есть.
- Нагревательный элемент оснащен высокотемпературной проводкой для надежной работы.

– Максимальная температура на поверхности нагревателя достигается через 1-2 минуты после включения.

– Точность температуры может варьироваться в пределах  $\pm 10^{\circ}\text{C}$ .

Преимущества:

- Безопасность благодаря изоляции поверхности.
- Быстрый нагрев и длительный срок службы.
- Постоянная температура, высокая надежность и легкость в использовании.
- Высокая мощность для таких компактных размеров.



Рисунок 4.2 – Внешний вид нагревательного элемента *Talk-Satisfied 04873*

Этот *PTC* нагревательный элемент был выбран для изготовления термостата для сепаратора дисплеев благодаря его способности быстро и эффективно нагревать поверхность до постоянной температуры, что необходимо для равномерного нагрева и безопасной работы с чувствительными материалами, такими как стекло и дисплеи смартфонов. Компактные размеры, высокая мощность и простота в использовании делают его оптимальным решением для данного проекта.

**4.1.3** Общий обзор дисплея LCD1602. *LCD 1602 I2C* дисплей – это модуль для отображения текста, предназначенный для использования в различных электронных устройствах. Данный дисплей построен на базе технологии жидкокристаллического экрана (*LCD*) с подсветкой, что позволяет работать даже в условиях недостаточного освещения. Он поддерживает интерфейс *I2C*, который упрощает подключение и управление, используя всего два провода для передачи данных и питания. (рисунок 4.3).



Рисунок 4.3 – Внешний вид LCD1602 дисплея

Данный дисплей поддерживает интерфейсы *I2C* и *SPI*, что делает его универсальным для интеграции с различными микроконтроллерами, включая *Arduino* и *Raspberry Pi*.

Технические характеристики [16]:

- Тип дисплея: Жидкокристаллический (*LCD*).
- Размер дисплея: 0,91 дюйма.
- Разрешение: 16x2 символа (16 столбцов и 2 строки текста).
- Цвет отображения: белый
- Подсветка: Встроенная светодиодная подсветка, цвет которой может быть настроен.
- Контроллер дисплея: *HD44780*.
- Интерфейс: *I2C* (*SDA* и *SCL*) или *SPI*.
- Питание: 5В.

В проекте термостата *LCD 1602 I2C* используется для отображения текущей температуры, установленной цели, состояния нагревательных элементов и другой информации. Благодаря простоте интеграции через

интерфейс *I2C* и совместимости с *Arduino*, он идеально подходит для таких приложений [17]. Кроме того, для дисплеев подобного типа давно существует большое количество отлаженных и надежных библиотек. В данной работе я остановился в своем выборе на *LiquidCrystal\_I2C.h* [18]. Стоит упомянуть Подсветка дисплея обеспечивает хорошую читаемость даже в условиях низкой освещенности [19].

Высокая контрастность и читаемость на дисплее позволяют оператору быстро и точно отслеживать температурные изменения, что важно для точного контроля процесса отделения стекла от матрицы в телефонах.

**4.1.4** Общий обзор блока питания системы *PROCASE 300X P4 300W*. Данный блок питания, используется в данной курсовой работе для обеспечения стабильного электропитания нагревательных элементов системы. Мощность в 300 Вт позволяет поддерживать заданную температуру, необходимую для работы нагревателей, при этом блок обладает достаточной производительностью и надежностью для длительного непрерывного использования. Устройство поддерживает автоматическую регулировку мощности и защиту от перегрузок, что гарантирует безопасную эксплуатацию в условиях интенсивного потребления энергии нагревательными элементами проекта [20].

## **4.2 Функции и устройство разрабатываемого устройства**

Сепаратор дисплейных модулей представляет собой установку для обеспечения точного нагрева металлической пластины, необходимой для последующей сепарации дисплейных модулей. Конструктивно устройство включает металлическую пластину, на нижней стороне которой размещены два нагревательных элемента с положительным температурным коэффициентом (*PTC*). Такая конструкция обеспечивает равномерный и контролируемый нагрев рабочей поверхности, что особенно важно при работе с чувствительными материалами. Нагревательные элементы активируются через реле, которое управляется цифровыми портами микроконтроллера. При этом каждое реле контролирует свой нагревательный элемент, что позволяет включать и отключать их независимо друг от друга. Возможность индивидуального управления каждым нагревателем повышает надежность системы и предотвращает перегрев отдельных участков пластины.

Для управления работой устройства предусмотрена удобная система из трех кнопок, каждая из которых выполняет свою определённую функцию. Первая кнопка отвечает за активацию режима нагрева. Пока пользователь не нажмёт эту кнопку, нагревательные элементы остаются выключенными, что

является дополнительной мерой безопасности. Таким образом, случайное срабатывание системы исключается, что защищает как оператора, так и устройство от потенциально опасных ситуаций. Вторая кнопка предназначена для увеличения целевой температуры. При каждом нажатии значение целевой температуры повышается на фиксированное значение, позволяя постепенно достигать необходимого уровня нагрева. Это позволяет оператору гибко подбирать температурный режим для работы с различными типами дисплеев, которые могут требовать разной степени нагрева. Максимальная температура, которую можно установить с помощью этой кнопки, составляет 125 °С. Третья кнопка, напротив, используется для понижения целевой температуры. При каждом нажатии температура уменьшается на заданный шаг, вплоть до минимально возможного значения в 20 °С. Подобная система управления позволяет пользователю полностью контролировать процесс нагрева и корректировать его на любом этапе работы. Такая гибкость управления температурой особенно важна при обработке модулей, которые отличаются по типу клеевого слоя или другим конструктивным особенностям.

Для удобства контроля за процессом работы сепаратор оснащён LCD-экраном, на который выводится вся актуальная информация о состоянии устройства. В реальном времени на экране отображаются два ключевых параметра: текущая температура металлической пластины и целевая температура, установленная пользователем. Данная функция индикации позволяет оператору непрерывно следить за процессом работы и оперативно вносить изменения при необходимости. Наличие визуальной индикации делает эксплуатацию устройства более удобной и безопасной. Если фактическая температура отличается от заданной, оператор может принять меры по её корректировке, используя кнопки управления. Это помогает избежать ситуаций, когда температура случайно выходит за пределы допустимых значений, что могло бы привести к повреждению дисплейного модуля.

Подключение сепаратора к сети электропитания осуществляется через стандартный силовой кабель, который подсоединяется к блоку питания. Работа устройства возможна при напряжении 220 В, что делает его удобным для использования в большинстве мастерских и производственных помещений. Встроенный блок питания обеспечивает стабильность работы и защищает устройство от скачков напряжения, предотвращая возможные поломки из-за перегрузок. Это повышает надёжность и долговечность работы сепаратора, делая его устойчивым к нестабильному напряжению в сети. С учётом всех вышеперечисленных особенностей устройство является надёжным и безопасным инструментом для работы с дисплейными модулями,

предлагая пользователю гибкость управления процессом нагрева и точный контроль температуры.

## 5 АРХИТЕКТУРА РАЗРАБАТЫВАЕМОЙ ПРОГРАММЫ

### 5.1 Общая структура программы

Программа предоставляет возможность управления температурой нагревательной установки с применением микроконтроллера и *LCD*-экрана, предназначенного для вывода данных. Главной функцией программы является поддержание заданной пользователем температуры с использованием двух нагревателей, управляемых через цифровые реле. Основные функции программы включают установку целевой температуры с использованием кнопок и отображение текущей и целевой температуры на *LCD*-экране. Программа состоит из нескольких блоков, обеспечивающих работу системы контроля температуры, среди которых инициализация оборудования, считывание данных с температурных датчиков, управление включением и выключением нагревателей, а также отображение данных на экране.

### 5.2 Описание функциональной схемы алгоритма

Алгоритм работы программы представлен в приложении В включает следующие функциональные этапы:

**5.2.1** Инициализация оборудования. На этапе *setup()* производится инициализация входных и выходных пинов, а также *LCD*-экрана, который используется для отображения текущей температуры нагревательной поверхности и целевой температуры, устанавливаемой пользователем. Устанавливаются начальные значения температуры, статус нагревателя (по умолчанию выключен) и значение гистерезиса.

**5.2.2** Чтение состояния кнопок. В основном цикле *loop()* программа регулярно проверяет состояние кнопок. Каждая из кнопок выполняет свою функцию:

- Кнопка увеличения температуры при нажатии увеличивает целевую температуру на фиксированное значение (5°C), если значение меньше 125°C.
- Кнопка уменьшения температуры: при нажатии уменьшает целевую температуру на 5°C, если значение больше 25°C.
- Кнопка запуска нагрева: активирует и деактивирует нагреватели при каждом нажатии, изменяя состояние переменной *heatingActive*.

**5.2.3** Чтение данных с температурных датчиков. Показания с двух датчиков температуры (*LM35* и *LM135*) регулярно считываются с аналоговых

пинов и усредняются для получения текущего значения температуры. Затем это значение используется для оценки, требуется ли корректировка работы нагревателей.

**5.2.4 Управление нагревателями.** В случае, если система нагрева активирована (*heatingActive = true*), программа сравнивает текущее значение температуры с целевой. Если текущая температура меньше целевой минус гистерезис, нагреватели включаются, иначе они выключаются. Это управление осуществляется через цифровые порты, подключенные к реле.

**5.2.5 Отображение данных на LCD-экране.** Программа выводит на экран текущее значение температуры и целевое значение температуры, установленное пользователем. Это позволяет оперативно отслеживать температурный режим и корректировать его в зависимости от требований.

### **5.3 Описание блок-схемы алгоритма**

Блок-схема алгоритма программы состоит из следующих основных компонентов:

1. Старт программы. Инициализация всех подключенных компонентов и *LCD*-экрана. Представлена в приложении Г на рисунке 1.
2. Основной цикл программы. В данном цикле, представленном блок схемой в графическом приложении Г на рисунке 2 осуществляется считывание состояний кнопок и показаний с датчиков и управление состоянием системы на основании полученных данных и состояний. Из данного цикла происходит вызов всех остальных функций программного кода.
3. Валидация данных. Представлена в графическом приложении Г на рисунке 3. Производит проверку показаний датчиков в, в случае ошибочных данных, аварийную остановку программы и вывод на дисплей сообщения об ошибке и необходимости ее исправить.
4. Отображение данных на *LCD*-экране. Представлено в приложении Г на рисунке 1. Программа обновляет значения текущей и целевой температур на *LCD*-экране для контроля пользователем.

### **5.4 Описание интерфейса разрабатываемого устройства**

Интерфейс пользователя сепаратора представлен двумя элементами: панелью кнопок и экраном *LCD 1602 I2C*, представляющим графический интерфейс пользователя.



**5.4.1** Описание панели кнопок разрабатываемого устройства. Интерактивный интерфейс разрабатываемого сепаратора, представлен тремя кнопками:

1. Кнопка запуска поддержания постоянной температуры. Представлена в приложении Д на рисунке 1 слева. При нажатии на данную кнопку, если процесс поддержания постоянной температуры был запущен, он остановится. В случае нажатия кнопки при неактивном процессе поддержания постоянной температуры процесс запустится: разрабатываемое устройство будет поддерживать температуру нагревательной пластины равной температуре, обозначенной на графическом интерфейсе как «*Target temperature*», в пределах гистерезиса. Значение гистерезиса задается программно. Активность нагревательных элементов, осуществляющуюся только в режиме поддержания постоянной температуры, можно отследить по двум светодиодам, расположенным на плате рядом с нагревательной пластиной.

2. Кнопка понижения целевой температуры. Представлена в приложении Д на рисунке 1 по центру. При нажатии значение «*Target temperature*», отображаемое на графическом интерфейсе разрабатываемого устройства, уменьшается на 5°C. Минимальное значение «*Target temperature*», которое может быть достигнуто таким образом, составляет 25°C и так же, как и интервал уменьшения температуры, задается программно.

3. Кнопка повышения целевой температуры. Представлена в приложении Д на рисунке 1 справа. При нажатии значение «*Target temperature*», отображаемое на графическом интерфейсе разрабатываемого устройства, повышается на 5°C. Максимальное значение «*Target temperature*», которое может быть достигнуто таким образом, составляет 125°C и так же, как и интервал увеличения температуры, задается программно.

**5.4.2** Описание графического интерфейса разрабатываемого устройства. Графический интерфейс разрабатываемого устройства может отображать следующие наборы информации:

1. Стандартное отображение информации о состоянии сепаратора. Представлено на рисунке 3 в приложении Д В данном состоянии на дисплей разрабатываемого устройства выводится информация о значении температуры, которая будет поддерживаться устройством в случае запуска процесса поддержания постоянной температуры, обозначенной как «*Target temp*», и о значении текущей температуры нагревательной пластины, полученной как среднее арифметическое показаний, полученных с температурных датчиков, обозначенном как «*Current temp*».

2. В случае получения от правого (если смотреть на нагревательную панель со стороны микроконтроллера) датчика значения, выходящего за рамки интервала от 0°C до 135°C, значения которого задаются программно. На графическом интерфейсе разрабатываемого устройства появится следующая надпись: «Right sensor err/Fix and restart». После чего микроконтроллер приостановит свою работу и возобновит ее в прежнем режиме только при перезапуске. Вывод ошибки представлен в приложении Д на рисунке 6.

3. В случае получения от левого (если смотреть на нагревательную панель со стороны микроконтроллера) датчика значения, выходящего за рамки интервала от 0°C до 135°C, значения которого задаются программно. На графическом интерфейсе разрабатываемого устройства появится следующая надпись: «Left sensor err/Fix and restart». После чего микроконтроллер приостановит свою работу и возобновит ее в прежнем режиме только при перезапуске. Вывод ошибки представлен в приложении Д на рисунке 5.

4. В случае обнаружения разницы в показаниях температурных датчиков более чем на 5 °C на графическом интерфейсе разрабатываемого устройства появится следующая надпись: «Some sensor err/Fix and restart». После чего микроконтроллер приостановит свою работу и возобновит ее в прежнем режиме только при перезапуске. Критическая разница в значениях температурных датчиков определяется программно. Вывод ошибки представлен в приложении Д на рисунке 4.

## 5.5 Обработка ошибок

В данном проекте термостата предусмотрена система обработки ошибок, которая направлена на обеспечение надёжной и безопасной работы устройства даже в случае возникновения сбоев в работе датчиков температуры. Реализация системы защиты от ошибок играет ключевую роль, так как отказ в работе датчика может привести к перегреву металлической пластины и повреждению обрабатываемых деталей. Основной механизм обработки ошибок сосредоточен в функции под названием *validation*, которая отвечает за проверку корректности данных, поступающих от двух датчиков температуры. Если в ходе проверки выявляются отклонения, система автоматически предпринимает соответствующие действия для устранения угрозы и информирования пользователя о проблеме.

Первым этапом проверки данных является сравнение показаний двух датчиков температуры. В нормальном режиме работы их значения должны быть максимально близки друг к другу, поскольку оба датчика измеряют температуру одной и той же поверхности. Однако, если разница между ними превышает заранее установленное значение, называемое *ETD (Error Tolerance*

*for Difference*), система интерпретирует это как сигнал о некорректной работе одного из датчиков. Такая ситуация может возникнуть по нескольким причинам: от механического повреждения датчика до сбоев в передаче сигнала. В ответ на обнаруженное отклонение на экран *LCD* выводится сообщение об ошибке с явным указанием необходимости исправить проблему и перезапустить устройство.

Вторым этапом проверки корректности работы системы является контроль значений температуры, поступающих от каждого из датчиков. Для этого используется параметр *ETV (Error Tolerance for Value)*, который определяет максимально допустимую температуру, превышение которой считается ошибкой. Если хотя бы один из датчиков регистрирует температуру, выходящую за допустимый диапазон, система немедленно реагирует. В этом случае на экран *LCD* выводится уведомление, в котором указывается конкретный источник проблемы — левый или правый датчик. Подобно первому этапу, нагревательные элементы в этом случае также немедленно отключаются.

Если хотя бы один из описанных выше типов ошибок обнаружен, программа завершает свою работу с помощью вызова функции *abort()*. Эта функция предназначена для немедленного прекращения выполнения программы и перехода системы в "безопасное состояние". После срабатывания функции *abort()* все процессы останавливаются, что гарантирует, что термостат не будет продолжать нагревать пластину или пытаться выполнять другие операции до устранения ошибки.

Кроме непосредственного контроля датчиков температуры, система предусматривает отключение нагревательных элементов в случае возникновения любой ошибки. В частности, оба нагревательных элемента, управляемые через выводы *heaterPin1* и *heaterPin2*, деактивируются, даже если они находились в активном состоянии. Таким образом, независимо от типа ошибки (отказ датчика, превышение температуры или другие сбои) система всегда выключает нагревательные элементы, предотвращая возможность перегрева рабочей поверхности. Это особенно важно, поскольку даже кратковременный перегрев может повлиять на срок службы устройства и качество работы с обрабатываемыми материалами.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта был разработан термостат для сепаратора дисплейных модулей на базе платформы *Arduino Uno*. Устройство предназначено для контроля и поддержания температуры нагревательной поверхности, используемой при разделении стекла и матрицы дисплея. В рамках проекта были успешно решены задачи выбора аппаратных компонентов, разработки алгоритмов управления температурой, программирования микроконтроллера и сборки рабочей системы.

Для реализации устройства были выбраны следующие компоненты: плата *Arduino Uno*, реле для управления нагревательным элементом, *PTC*-нагреватели, температурные датчики *LM35*, а также другие компоненты. Использование *PTC*-нагревателя обеспечило стабильное и безопасное повышение температуры до заданного уровня, а применение двух датчиков *LM35* позволило отслеживать температурные изменения с высокой точностью.

Программное обеспечение для микроконтроллера было разработано в среде *Arduino IDE*. Основные функции программы включают считывание данных с температурных датчиков, обработку пользовательских команд через кнопки управления, а также активацию или деактивацию нагревателя в зависимости от заданных условий. Пользователь может с помощью кнопок устанавливать целевую температуру и запускать процесс нагрева, что делает устройство интуитивно понятным в эксплуатации.

В проекте была также реализована защита от ошибок. Например, при неисправности одного из датчиков система автоматически переходит в безопасный режим, отключая нагревательный элемент. Это существенно повышает надёжность работы устройства и минимизирует риски.

Тестирование устройства подтвердило его функциональность и соответствие заявленным требованиям. Разработанный термостат поддерживает заданный температурный диапазон с высокой точностью, что позволяет успешно использовать его в процессе отделения стекла от матрицы в дисплейных модулях.

Итогом курсового проекта стало создание прототипа термостата, который представляет собой надёжное, компактное и энергоэффективное решение. Работа над проектом позволила освоить основы программирования микроконтроллеров, проектирования электронных схем и разработки алгоритмов управления. Разработанный термостат может быть доработан для более сложных применений или адаптирован под другие задачи в сфере терморегуляции.

## СПИСОК ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

- [1] Стандарт Предприятия. Дипломные проекты (работы). Общие требования / А. Т. Доманов, Н. И. Сорока. – Минск БГУИР, 2023 [Электронный ресурс]. – Режим доступа: [https://www.bsuir.by/m/12\\_100229\\_1\\_121330.pdf](https://www.bsuir.by/m/12_100229_1_121330.pdf) – Дата доступа: 30.09.2024.
- [2] Arduino UNO R3 Technical Reference [Электронный ресурс]. – Режим доступа: <https://docs.arduino.cc/hardware/uno-rev3/>. – Дата доступа: 08.11.2024.
- [3] Плата Arduino Mega 2560 R3 [Электронный ресурс]. – Режим доступа: <https://protronics.ru/product/info?product=2?product=2>. – Дата доступа: 27.09.2024
- [4] The History of Arduino [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/en/about/>. – Дата доступа: 08.12.2023.
- [5] Общий обзор Arduino Nano [Электронный ресурс]. – Режим доступа: [https://3drob.ru/stati/pro\\_arduino/platy\\_3/arduino\\_nano/](https://3drob.ru/stati/pro_arduino/platy_3/arduino_nano/). – Дата доступа: 05.10.2024.
- [6] Evolution of Arduino: From Classroom Project to Global Phenomenon [Электронный ресурс]. – Режим доступа: <https://spectrum.ieee.org/the-making-of-arduino>. – Дата доступа: 15.02.2024.
- [7] Arduino IDE Programming Guide [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/reference/en/>. – Дата доступа: 09.10.2023.
- [8] Arduino bootloader documentation [Электронный ресурс]. – Режим доступа: <https://docs.arduino.cc/retired/hacking/software/Bootloader/> – Дата доступа: 27.09.2024
- [9] Arduino Debugging Tools [Электронный ресурс]. – Режим доступа: <https://docs.arduino.cc/software/ide-v2/tutorials/debugging/>. – Дата доступа: 08.12.2023.
- [10] LM35DZ Datasheet [Электронный ресурс]. – Режим доступа: <https://static.chipdip.ru/lib/361/DOC026361288.pdf>. – Дата доступа: 08.12.2024.
- [11] Chipdip. LM35DZ/NOPB, Прецизионный датчик температуры [Электронный ресурс]. – Режим доступа: <https://www.chipdip.ru/product/lm35dz> – Дата доступа: 12.10.2024.
- [12] LM35 Precision Centigrade Temperature Sensors [Электронный ресурс]. – Режим доступа: <https://static.chipdip.ru/lib/142/DOC000142743.pdf/>. – Дата доступа: 05.10.2024.
- [13] LM35 Sensor Datasheet [Электронный ресурс]. – Режим доступа: <https://www.alldatasheet.com/datasheet-pdf/view/1188696/TGS/LM35.html>. – Дата доступа: 14.10.2024.

[14] Alibaba Нагревательный элемент Mini PTC [Электронный ресурс]. – Режим доступа: <https://russian.alibaba.com/p-detail/12V-1600321449572.html>. – Дата доступа: 08.12.2024.

[15] 220 В 12 В PTC нагреватели нагревательный элемент *Talk-Satisfied 04873* [Электронный ресурс]. – Режим доступа: <https://www.bestsuppliers.com/products/458x8zmc2tk/220v-12v-ptc-heaters-heating-element-for-mini-incubator-hair-dryer-curlers-heater>. – Дата доступа: 05.10.2024.

[16] IARDUINO Символьный дисплей LCD1602 [Электронный ресурс]. – Режим доступа: <https://iarduino.ru/shop/displays/lcd1602-lcd-blue-golubaya-podsvetka-16x2-1602.html>. – Дата доступа: 19.11.2024

[17] I2C-bus specification and user manual [Электронный ресурс]. – Режим доступа: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. – Дата доступа: 30.09.2024.

[18] LiquidCrystal Library Reference [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/en/Reference/LiquidCrystal>. – Дата доступа: 08.10.2024.

[19] LCD1602 HD44780 Display Module Documentation [Электронный ресурс]. – Режим доступа: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>. – Дата доступа: 05.10.2024

[20] Блок питания 300W Procase GAF300 [Электронный ресурс]. – Режим доступа: [https://mmg.by/shop/w/Item/300w\\_procase\\_gaf300/](https://mmg.by/shop/w/Item/300w_procase_gaf300/). – Дата доступа: 30.10.2024.

Расширить список источников. Указать все источники картинок в работе

# ПРИЛОЖЕНИЕ А

## (обязательное)

### Справка о проверке на заимствования

Антиплагиат онлайн

Получить текст по URL

или

из файла

4.2 Функции и устройство разрабатываемого устройства

Сепаратор дисплейных модулей представляет собой установку для обеспечения точного нагрева металлической пластины, необходимой для последующей сепарации дисплейных модулей. Конструктивно устройство включает два нагревательных элемента с положительным температурным коэффициентом (ПТС), которые размещены на нижней стороне пластины, обеспечивая равномерный и контролируемый нагрев рабочей поверхности. Нагревательные элементы активируются независимо друг от друга через реле, управляемое с цифровых портов микроконтроллера. Данный подход позволяет эффективно регулировать процесс нагрева, предоставляя пользователю возможность гибкого управления температурой, что особенно важно при работе с материалами, требующими деликатного нагрева.

Для управления работой нагревателей предусмотрены три кнопки. Первая из них выполняет функцию разрешения запуска нагревательного процесса, при этом без активации этой кнопки нагревательные элементы остаются выключенными, что повышает уровень безопасности использования сепаратора. Вторая кнопка служит для пошагового увеличения целевой температуры, при каждом нажатии повышая её на фиксированное значение. Это позволяет пользователю устанавливать оптимальный температурный режим вплоть до максимального значения 120 °С. Третья кнопка, напротив, снижает целевую температуру с таким же фиксированным шагом до минимального предела в 20 °С, предоставляя пользователю полный контроль над параметрами нагрева в зависимости от требований обрабатываемого материала.

Развернуть...

Подсветка

Очистить

Символов: 9804 Слов: 1177

Исключить

страницу

https://site.ru/some-page/...

Проверить

Заказать уникализацию за 40 руб.

Заказать рерайт за 692 руб.

Рисунок 1 – Результат проверки на заимствование

Уникальность текста: <b>74%</b>	Точность текста: <b>3.7%</b>	Грамматических ошибок: <b>1</b>	Водность текста: <b>21%</b>		
Авторство принадлежит	Совпадений	<a href="#">подсветить все</a>			
<a href="https://zont.online/wp-content/uploads/2022/11/Tekhnicheskaya-dokumentaciya-ZONT-CLIMATICy4.pdf">https://zont.online/wp-content/uploads/2022/11/Tekhnicheskaya-dokumentaciya-ZONT-CLIMATICy4.pdf</a>	2.4%	<a href="#">подсветить</a>			
<a href="https://soltau.ru/index.php/arduino/item/371-interfejs-12c-i-arduino">https://soltau.ru/index.php/arduino/item/371-interfejs-12c-i-arduino</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://ru.wikipedia.org/wiki/%D0%96%D0%B8%D0%B4%D0%BA%D0%BE%D0%BA%D1%80%D0%B8%D1%81%D1%82%...">https://ru.wikipedia.org/wiki/%D0%96%D0%B8%D0%B4%D0%BA%D0%BE%D0%BA%D1%80%D0%B8%D1%81%D1%82%...</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://support.microline.ru/index.php/%D0%A2%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%...">https://support.microline.ru/index.php/%D0%A2%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%...</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://otvet.mail.ru/question/202431866">https://otvet.mail.ru/question/202431866</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://kubsau.ru/upload/iblock/4cf/4cf3ad7d6ae3c54ce2f48e66bae79507.pdf">https://kubsau.ru/upload/iblock/4cf/4cf3ad7d6ae3c54ce2f48e66bae79507.pdf</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://www.snijp.ru/assets/images/resources/5816/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%2...">https://www.snijp.ru/assets/images/resources/5816/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%2...</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://cae-club.ru/tooltips">https://cae-club.ru/tooltips</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://www.mgau.ru/students/umk/files/umk09.02.07/%D0%9F%D0%9C%D0%9F%D0%9C.03%D0%A1%D0%BE%...">https://www.mgau.ru/students/umk/files/umk09.02.07/%D0%9F%D0%9C%D0%9F%D0%9C.03%D0%A1%D0%BE%...</a>	1.6%	<a href="#">подсветить</a>			
<a href="https://www.vstu.ru/upload/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%20%D0%9A%D0%BE%D0%...">https://www.vstu.ru/upload/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%20%D0%9A%D0%BE%D0%...</a>	1.6%	<a href="#">подсветить</a>			
<a href="#">Показать все</a>					
<div> <div>Проверено шинглов: 123 / 123</div> <div>Уникальных: 92</div> <div>Неуникальных: 31</div> </div>					

Рисунок 2 – Список источников

41

## ПРИЛОЖЕНИЕ Б

### (обязательное)

#### Листинг программного кода

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

const int buttonPinUp = 3;
const int buttonPinDown = 8;
const int buttonPinStart = 2;

const int heaterPin1 = 5;
const int heaterPin2 = 6;

const int tempSensor35DzOldPin = A3;
const int tempSensor35DzNewPin = A0;

const unsigned long errorTemperatureDelta = 5;
const unsigned long ETD = errorTemperatureDelta * 1023 / 500;
const unsigned long errorTemperatureValue = 105;
const unsigned long ETV = errorTemperatureValue * 1023 / 500;

bool heatingActive = false;
bool keepConstantTemperature = false;

unsigned int targetTemp = 30;
unsigned int currentTemp = 0;
unsigned int hysteresis = 1;

unsigned long lastUpdate = 0;
const int updateInterval = 1000;

LiquidCrystal_I2C LCD(0x27, 16, 2);

void setup() {
    pinMode(13, OUTPUT);

    pinMode(buttonPinUp, INPUT);
    pinMode(buttonPinDown, INPUT);
    pinMode(buttonPinStart, INPUT);
    pinMode(heaterPin1, OUTPUT);
    pinMode(heaterPin2, OUTPUT);
    digitalWrite(heaterPin1, HIGH);
    digitalWrite(heaterPin2, HIGH);

    Serial.begin(9600);
    Serial.println("setup ready");
}
```



```

    LCD.init();
    LCD.backlight();

    LCD.setCursor(1, 0);
    LCD.print("Welcome");

    LCD.setCursor(1, 1);
    LCD.print("back");
    LCD.display();
    delay(3000);
    LCD.clear();
}

void loop() {
    int upStat = digitalRead(buttonPinUp);
    int downStat = digitalRead(buttonPinDown);
    int startStat = digitalRead(buttonPinStart);

    unsigned          int          sensor35OldValue          =
    analogRead(tempSensor35DzOldPin);
    unsigned          int          sensor35NewValue          =
    analogRead(tempSensor35DzNewPin);

    validation(sensor35OldValue, sensor35NewValue);

    int temp35Old = sensor35OldValue * 500 / 1024;
    int temp35New = sensor35NewValue * 500 / 1024;
    currentTemp = (temp35Old + temp35New) / 2;

    if (upStat == LOW && targetTemp < 125) {
        targetTemp += 5;
        delay(500);
    }

    if (downStat == LOW && targetTemp > 25) {
        targetTemp -= 5;
        delay(500);
    }

    if (startStat == LOW) {
        keepConstantTemperature = !keepConstantTemperature;
        delay(500);
    }

    if (millis() - lastUpdate >= updateInterval)
    {

```

```

        lastUpdate = millis();
        updateScreen(sensor35OldValue, temp35Old, sensor35NewValue,
temp35New, currentTemp);
    }

    if (keepConstantTemperature)
    {
        if (currentTemp < targetTemp - hysteresis) {
            heatingActive = true;
            digitalWrite(heaterPin1, LOW);
            digitalWrite(heaterPin2, LOW);
            delay(500);
        } else if (currentTemp > targetTemp + hysteresis) {
            heatingActive = false;
            digitalWrite(heaterPin1, HIGH);
            digitalWrite(heaterPin2, HIGH);
            delay(500);
        }
    }
}

void updateScreen(int sensor35OldValue, int temp35Old, int
sensor35NewValue, int temp35New, int currentTemp)
{
    LCD.clear();
    LCD.setCursor(0, 0);
    LCD.print("Target temp ");
    LCD.print(targetTemp);

    LCD.setCursor(0, 1);
    LCD.print("Current ");

    LCD.print(currentTemp);
    LCD.print(" ");
    LCD.print(temp35Old);
    LCD.print(" ");
    LCD.print(temp35New);
}

void validation(int sensor35OldValue, int sensor35NewValue)
{
    bool readingIsValid = true;
    if (abs(sensor35NewValue - sensor35OldValue) > ETD)
    {
        LCD.clear();
        LCD.setCursor(0, 0);
        LCD.print("Some sensor err!");
    }
}

```

```

    LCD.setCursor(0, 1);
    LCD.print("Fix and restart");

    readingIisValid = false;
}

if (sensor35OldValue > ETV)
{
    LCD.clear();
    LCD.setCursor(0, 0);
    LCD.print("Left sensor err!");
    LCD.setCursor(0, 1);
    LCD.print("Fix and restart");

    digitalWrite(heaterPin1, HIGH);
    digitalWrite(heaterPin2, HIGH);

    readingIisValid = false;
}

if (sensor35NewValue > ETV)
{
    LCD.clear();
    LCD.setCursor(0, 0);
    LCD.print("Right sensor err!");
    LCD.setCursor(0, 1);
    LCD.print("Fix and restart");

    readingIisValid = false;
}

if(!readingIisValid)
{
    if (heatingActive)
    {
        digitalWrite(heaterPin1, false);
        digitalWrite(heaterPin2, false);
    }
    abort();
}}

```

**ПРИЛОЖЕНИЕ В**  
**(обязательное)**  
**Функциональная схема алгоритма, реализующего программное**  
**средство**

**ПРИЛОЖЕНИЕ Г**  
**(обязательное)**

**Блок схема алгоритма, реализующего программное средство**

**ПРИЛОЖЕНИЕ Д**  
**(обязательное)**  
**Графический интерфейс пользователя**

**ПРИЛОЖЕНИЕ Д**  
**(обязательное)**  
**Схема разрабатываемого устройства**

**ПРИЛОЖЕНИЕ Е**  
**(обязательное)**  
**Ведомость документов курсового проекта**