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:*

И.О. Заведующего кафедрой информатики

\_\_\_\_\_\_\_\_\_\_\_\_\_С. И. Сиротко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

**СИСТЕМА ОПРЕДЕЛЕНИЯ ГОРЕНИЯ ЗАГОТОВКИ В ПРОЦЕССЕ ЛАЗЕРНОЙ РЕЗКИ НА БАЗЕ ESP32**

БГУИР КП 1-40 04 01 015 ПЗ

Студент Сенько Н. С.

Руководитель Калиновская А. А.

Нормоконтролер Калиновская А. А.

Минск 2024

**CОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc181262899)

[1 АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ 6](#_Toc181262900)

[1.1 Понятие микроконтроллера 6](#_Toc181262901)

[1.2 Платформа Arduino 6](#_Toc181262902)

[1.3 История развития плат Arduino 9](#_Toc181262903)

[1.4 Датчик пламени пятиканальный 13](#_Toc181262904)

[1.5 Обоснование выбора вычислительной системы 14](#_Toc181262905)

[2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 15](#_Toc181262906)

[2.1 Среда разработки (IDE) 15](#_Toc181262907)

[2.2 PlatformIO история, версии, достоинства 16](#_Toc181262908)

[2.3 Процесс прошивки Arduino 17](#_Toc181262909)

[2.4 Язык программирования 17](#_Toc181262910)

[2.5 Обоснование выбора программной платформы 18](#_Toc181262911)

[3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА 20](#_Toc181262912)

[3.1 Углубленный анализ инфракрасных датчиков пламени 20](#_Toc181262913)

[3.2 Калибровка датчиков 21](#_Toc181262914)

[3.3 Пользовательский интерфейс 21](#_Toc181262915)

[4 ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ПРОГРАММЫ 23](#_Toc181262916)

[4.1 Обоснование и описание функциональных возможностей 23](#_Toc181262917)

[5 АРХИТЕКТУРА РАЗРАБАТЫВАЕМОЙ ПРОГРАММЫ 25](#_Toc181262918)

[5.1 Подключение устройств к Arduino 26](#_Toc181262919)

[5.2 Общая структура программы 28](#_Toc181262920)

[5.3 Описание функциональной схемы программы 29](#_Toc181262921)

[5.4 Описание блок-схемы алгоритма программы 30](#_Toc181262922)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 32](#_Toc181262923)

# ВВЕДЕНИЕ

Курсовая работа посвящена разработке системы определения горения заготовки в процессе лазерной резки на базе микроконтроллера Arduino. Основная задача — подобрать и проанализировать данные с инфракрасных датчиков пламени для определения наличия горения, а также обработать возможные неисправности датчиков и краевые случаи. Система должна обеспечивать надежное и точное определение горения в режиме реального времени, что критически важно для обеспечения безопасности и эффективности процесса лазерной резки.

Актуальность курсовой работы обусловлена несколькими факторами. Во-первых, минимизация человеческого фактора и возможность использования в опасных или труднодоступных условиях повышает безопасность производственного процесса. Во-вторых, автоматическая система способна обеспечить повышение надежности и точности измерений в реальном времени, что критически важно для оптимизации процесса лазерной резки. В-третьих, реализация удаленного мониторинга и управления позволяет оперативно реагировать на изменения в процессе и предотвращать потенциальные проблемы.

Цель проекта – разработка и реализация автоматической системы, способной точно и стабильно определять наличие горения, и интеграция с промышленными интерфейсами для эффективного управления и передачи данных.

Перечень планируемых задач:

– Подобрать и настроить инфракрасные датчики пламени.

– Интегрироваться с интерфейсами передачи данных.

– Протестировать и отладить систему.

Детальная постановка задачи:

– Изучить техническую документацию используемых датчиков и их интерфейсов.

– Исследовать функциональные возможности Arduino для выполнения задачи.

– Реализовать анализ данных для определения наличия горения.

– Провести тестирование системы, включая проверку точности анализа.

– Проверить корректность передачи данных и работу системы управления.

Реализация анализа данных для определения наличия горения является ключевым этапом работы. Это включает в себя разработку алгоритмов обработки сигналов с датчиков, фильтрацию шумов, определение пороговых значений для детекции горения и учет возможных краевых случаев. Важно также предусмотреть механизмы самодиагностики системы для выявления неисправностей датчиков или аномальных показаний.

# 1 АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

## Понятие микроконтроллера

Появление первых микроконтроллеров стало важной вехой в истории развития компьютерной техники. Именно с их появлением началась новая эра, связанная с массовым применением автоматизации в различных сферах управления. Микроконтроллер представляет собой сложное устройство, в котором на одном кристалле объединены центральный процессор, память и периферийные устройства. Эти компоненты позволяют микроконтроллерам выполнять разнообразные задачи по обработке данных и управлению внешними системами. Первый патент на однокристальную микроЭВМ был выдан в 1971 году инженерам Майклу Кокрэну и Гэри Буну, сотрудникам американской Texas Instruments. Именно они предложили на одном кристалле разместить не только процессор, но и память с устройствами ввода-вывода.

Одной из проблем работы с микроконтроллерами является необходимость знания схемотехники и устройства конкретного процессора и умения программировать на ассемблере для данного процессора. Также для разработки необходимы программаторы и отладчики, что долгое время не позволяло любителям использовать микроконтроллеры в своих проектах. Однако на сегодняшний день появились устройства, позволяющие работать с микроконтроллерами без наличия серьезной материальной базы и знания многих предметов. Примером такого устройства является проект Arduino, разработанный в Италии в 2005 году.[2]

## Платформа Arduino

Устройства из семейства Arduino представляют собой наборы, состоящие из электронного блока и программного обеспечения. Электронный блок состоит из печатной платы с установленным микроконтроллером ATMega фирмы Atmel и набором элементов, необходимых для его работы. По сути электронный блок можно считать аналогом материнской платы современного компьютера. На нем имеются разъемы для подключения внешних устройств, а также разъем для связи с компьютером, по которому и осуществляется программирование микроконтроллера.

С 2008 года в компании-разработчике начался раскол, выразившийся в существовании двух независимых ветвей развития и продаж под одной торговой маркой: одна на сайте arduino.cc, другая на arduino.org. Докризисные изделия на обоих сайтах продаются под одинаковыми названиями. Набор новых изделий на сайтах различается. Также существует две ветви Arduino IDE, поддерживающие разный набор плат и библиотек. Одинаковые названия и пересекающиеся номера версий IDE вносят путаницу.

Под торговой маркой Arduino выпускается несколько плат с микроконтроллером и платы расширения (так называемые шилды). Большинство плат с микроконтроллером снабжены минимально необходимым набором обвязки для нормальной работы микроконтроллера (стабилизатор питания, кварцевый резонатор, цепочки сброса и т. п.).

Arduino и Arduino-совместимые платы спроектированы таким образом, чтобы их можно было при необходимости расширять, добавляя в устройство новые компоненты. Эти платы расширений подключаются к Arduino посредством установленных на них штыревых разъёмов. Существует ряд плат с унифицированным конструктивом, допускающим конструктивно жесткое соединение процессорной платы и плат расширения в стопку через штыревые линейки. Кроме того, выпускаются платы уменьшенных габаритов (например, Nano, Lilypad) и специальных конструктивов для задач робототехники. Независимыми производителями также выпускается большое количество всевозможных датчиков и исполнительных устройств, в той или иной степени совместимых с базовым конструктивом Arduino. На рисунке 1.1 представлена плата расширения Sensor Shield 4.

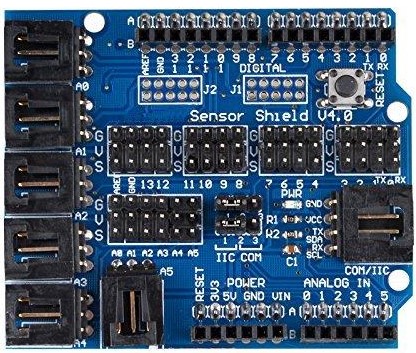


Рисунок 1.1 – Плата расширения Sensor Shield 4

Микроконтроллеры для Arduino отличаются наличием предварительно прошитого в них загрузчика (bootloader). С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов. Загрузчик соединяется с компьютером через интерфейс USB (если он есть на плате) или с помощью отдельного переходника UART-USB. Поддержка загрузчика встроена в Arduino IDE и выполняется в один щелчок мыши.

На случай затирания загрузчика или покупки микроконтроллера без загрузчика разработчики предоставляют возможность прошить загрузчик в микроконтроллер самостоятельно. Для этого в Arduino IDE встроена поддержка нескольких популярных дешевых программаторов, а большинство плат Arduino имеет штыревой разъем для внутрисхемного программирования (ICSP для AVR, JTAG для ARM).

В Arduino IDE от компании, базирующейся на сайте arduino.cc, встроена возможность создания своих программно-аппаратных платформ. Этой возможностью пользуются сторонние компании, добавляющие в Arduino IDE свои наборы плат и компиляторов-загрузчиков к ним. Компания на сайте arduino.org не поддерживает такую возможность.

В линейке устройств Arduino в основном применяются микроконтроллеры Atmel AVR: ATmega328, ATmega168, ATmega2560, ATmega32U4, ATTiny85 с частотой тактирования 16 или 8 МГц. В старых изделиях применялись ATmega8, ATmega1280 и другие. Есть также платы на процессоре ARM Cortex M. Сторонние разработчики портировали в Arduino поддержку популярного Wi-Fi микроконтроллера ESP8266. Теперь компилировать и загружать прошивку для ESP8266 со своими скетчами и поддержкой Wi-Fi можно прямо из Arduino IDE, получая одноплатную схему с поддержкой сети Wi-Fi.

Порты ввода-вывода микроконтроллеров оформлены в виде штыревых линеек. Никакого буферизирования, защиты, конвертации уровней или подтяжек, как правило, нет. Микроконтроллеры питаются от 5В или 3,3В, в зависимости от модели платы. Соответственно порты имеют такой же размах допустимых входных и выходных напряжений. Программисту доступны некоторые специальные возможности портов ввода-вывода микроконтроллеров, например, широтно-импульсная модуляция, аналогово-цифровой преобразователь, интерфейсы UART, SPI, I2C. Количество и возможности портов ввода-вывода определяются конкретным вариантом микропроцессорной платы.

Помимо портов на платах микроконтроллеров иногда устанавливается периферия в виде интерфейсов USB или Ethernet.

Сторонние производители выпускают широкую гамму датчиков и исполнительных устройств, подключаемых к Arduino. Например, гироскопы, компасы, манометры, гигрометры, термометры, релейные модули, индикаторы, клавиатуры и т. п. Всё это превращает Arduino в универсальное ядро системы, которое может быть сконфигурировано совершенно разнообразными способами.

Таким образом, можно выделить следующие преимущества платформы Arduino:

1 Программное обеспечение Arduino совместимо с операционными системами Windows, Macintosh OSX и Linux, в отличие от большинства аналогичных систем, работающих только в Windows.

2 Arduino предоставляет возможность как профессионалам, так и новичкам заниматься разработкой микропроцессорных устройств. Благодаря наличию готовых модулей и библиотек программ, создание функционирующих устройств становится гораздо проще даже для тех, кто не имеет опыта в электронике.

3 Семейство платформ Arduino включает различные модели, от компактных до более мощных, что позволяет выбрать наиболее подходящий вариант для конкретного проекта.

4 Для Arduino не нужен программатор. Всё сделано так, чтобы программирование Arduino для начинающих не составляло труда. Написанный код можно загрузить в микроконтроллер посредством USB-кабеля. Достигается это преимущество не каким-то встроенным уже заранее программатором, а специальной прошивкой – бутлоадером. Бутлоадер является специальной программой, которая запускается сразу после подключения и слушает, будут ли какие-то команды, прошивать ли кристалл, есть ли проекты Arduino или нет.

## История развития плат Arduino

В данном разделе будет рассмотрена история развития официальных плат Arduino по годам с перечислением их ключевых изменений.

Arduino Diecimila – первая версия платы Arduino, выпущенная в 2005 году. Она была разработана как простое и доступное средство для обучения программированию и электронике. Данная плата представлена на рисунке 1.2.



Рисунок 1.2 – Плата Arduino Diecimila

В 2006 году была представлена новая версия платы — Arduino Duemilanove, которая быстро завоевала популярность среди энтузиастов и разработчиков. Эта плата отличалась улучшенной совместимостью с различными устройствами и поддержкой нескольких типов микроконтроллеров, что значительно расширило её функциональность. Благодаря этим усовершенствованиям, Duemilanove стал основой для множества интересных и инновационных проектов в области электроники и программирования. Плата предоставила пользователям возможность легко интегрировать её в различные приложения, от простых до более сложных систем. Данная плата представлена на рисунке 1.3.

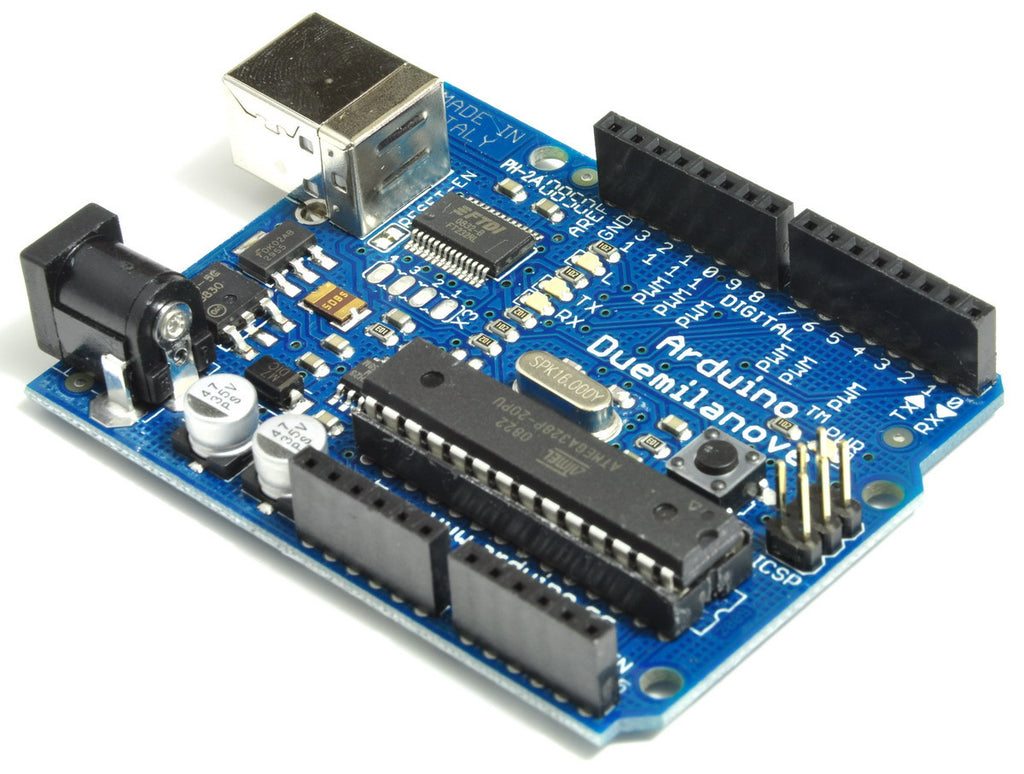


Рисунок 1.3 – Плата Arduino Duemilanove

Arduino Nano — это компактная плата, представленная в 2008 году, которая идеально подходит для работы в условиях ограниченного пространства. Благодаря своим минимальным размерам, она легко интегрируется в различные проекты, где важно экономить место. Nano использует микроконтроллеры ATmega168 или ATmega328, что обеспечивает ей хорошую производительность при малом энергопотреблении. Эта плата стала особенно популярной среди разработчиков, работающих над малыми проектами и прототипами, благодаря своей универсальности и простоте в использовании. Данная плата представлена на рисунке 1.4.



Рисунок 1.4 – Плата Arduino Nano

Arduino Uno — это плата, которая пришла на смену Duemilanove и была представлена в 2009 году. Она основана на микроконтроллере ATmega328, который обеспечивает высокую производительность и широкие возможности для разработчиков. Uno быстро стала стандартом для большинства проектов благодаря своей простоте в использовании и обширному сообществу. Эта плата поддерживает множество библиотек и примеров, что делает её идеальным выбором как для новичков, так и для опытных пользователей. Данная плата представлена на рисунке 1.5.

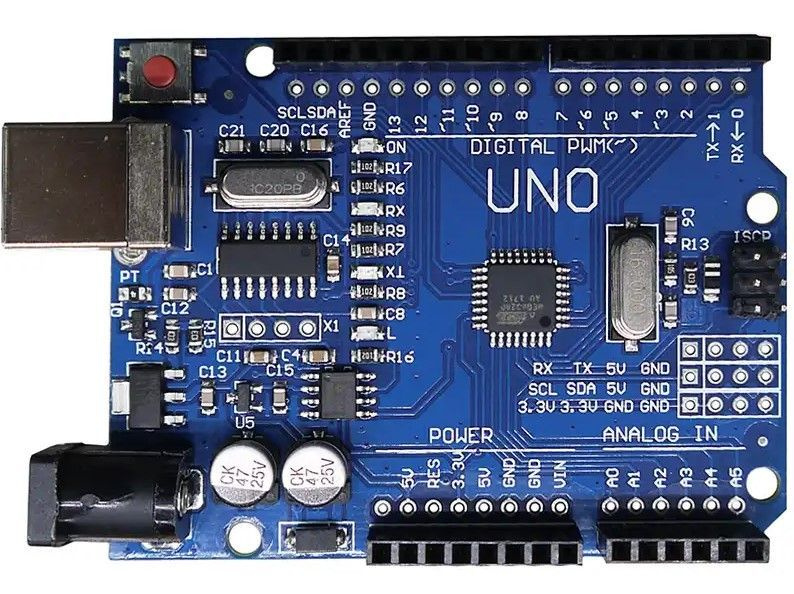


Рисунок 1.5 – Плата Arduino UNO

В 2010 году вышла новая плата – Arduino Mega с большим количеством входов/выходов (54 цифровых и 16 аналоговых). Она предназначена для более сложных проектов, требующих большей памяти и вычислительной мощности. Данная плата представлена на рисунке 1.6.



Рисунок 1.6 – Плата Arduino Mega 2560

Arduino Leonardo — это уникальная плата, представленная в 2011 году, которая выделяется своей способностью эмулировать USB-устройства, такие как клавиатуры и мыши. Эта функция значительно расширила возможности взаимодействия с компьютером, позволяя разработчикам создавать инновационные проекты, которые могут напрямую взаимодействовать с программным обеспечением на ПК. Благодаря использованию микроконтроллера ATmega32U4, Leonardo может отправлять данные через USB, что делает её отличным выбором для создания различных интерфейсов и управляемых устройств. Например, с помощью этой платы можно разработать проекты, которые автоматически вводят текст, управляют курсором или выполняют другие действия, как будто это делает обычная клавиатура или мышь. Данная плата представлена на рисунке 1.7.

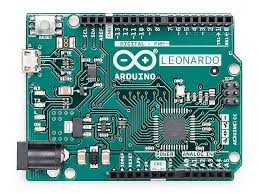


Рисунок 1.7 – Плата Arduino Leonardo

В 2013 году была представлена Arduino Due — первая плата в линейке Arduino, оснащённая 32-битным ARM Cortex-M3 процессором. Это значительное обновление обеспечило пользователям более высокую производительность и расширенные возможности по сравнению с предыдущими моделями. Arduino Due предлагает увеличенную скорость обработки данных и большую ёмкость памяти, что делает её идеальной для сложных проектов, требующих значительных вычислительных ресурсов. Плата поддерживает множество современных интерфейсов, включая USB, I2C и SPI, что позволяет легко интегрировать её в различные системы и устройства. Данная плата представлена на рисунке 1.8.



Рисунок 1.8 – Плата Arduino Due

Arduino MKR — это серия плат, ориентированных на Интернет вещей (IoT), представленная в 2015 году. Эти платы оснащены встроенными модулями для беспроводной связи, такими как Wi-Fi, LoRa и GSM, что делает их идеальными для разработки проектов, связанных с удалённым мониторингом и управлением. Серия MKR предлагает компактный и энергоэффективный дизайн, что позволяет легко интегрировать платы в различные устройства и системы. Каждая плата в этой серии поддерживает стандарты подключения, обеспечивая простоту в создании сетевых приложений. Данная плата представлена на рисунке 1.9.

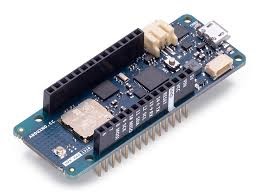


Рисунок 1.9 – Плата Arduino MKR

## Датчик пламени пятиканальный

Пятиканальный датчик пламени представляет собой устройство, предназначенное для обнаружения пламени в ближнем инфракрасном диапазоне. Он способен регистрировать пламя в диапазоне длин волн от 700 до 1100 нм и выдавать электрический сигнал (напряжение), который можно использовать для дальнейшей обработки и анализа. Внешний вид данного датчика представлен на рисунке 1.10.



Рисунок 1.10 – Пятиканальный инфракрасный датчик пламени

Этот модуль выполнен на одной печатной плате, что делает его компактным и удобным для интеграции в различные системы.

Аналоговое выходное напряжение датчика (на аналоговом выводе) обычно обеспечивает различную пропорцию аналогового выходного напряжения в зависимости от интенсивности пламени. Поэтому высокая интенсивность пламени соответствует высокому выходному напряжению и наоборот.

Затем компаратор получает тот же аналоговый сигнал. Затем он подает сигнал на цифровой выход (DO).

Принцип работы датчика заключается в том, что он использует фотодиоды, чувствительные к ближнему инфракрасному излучению. Когда пламя попадает в поле зрения датчика, фотодиоды регистрируют инфракрасное излучение и преобразуют его в электрический сигнал. Этот сигнал затем обрабатывается и выдается на аналоговый выход.

Особенности модуля:

1 Двоичная индикация – позволяет определить наличие или отсутствие алкоголя в окружающей среде.

2 Аналоговый выход – предоставляет информацию о концентрации алкоголя в виде изменяющегося аналогового напряжения.

3 Легкость использования – простое взаимодействие с Arduino и другими микроконтроллерами.

4 Наличие пяти каналов на одной плате – позволяет существенно упростить работу с модулем, по сравнению с пятью отдельными модулями.

## Обоснование выбора вычислительной системы

Основные преимущества выбранных для выполнения данной курсовой работы устройств:

– Arduino представляет собой готовый электронный модуль с установленным микроконтроллером и основным набором элементов. Эта платформа разработана с учетом возможности дальнейшего расширения, что дает возможность легко добавлять новые компоненты. Главным преимуществом является то, что разработчикам не нужно тратить время на создание законченных схем и модулей, что упрощает процесс разработки.

– Инфракрасные датчики пламени обеспечивают возможность обнаружения пламени в ближнем инфракрасном диапазоне (700-1100 нм). Использование нескольких датчиков позволяет корректно обработать ситуацию с выходом одного из датчиков из строя, что обеспечивает надежную и точную работу системы.

Таким образом, платформа Arduino и выбранные дополнительные устройства являются оптимальными для реализации данного курсового проекта благодаря следующим преимуществам.

# 2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В этом разделе будет подробно рассмотрена программная платформа, выбранная для выполнения курсового проекта. Будет проанализирована как среда разработки, так и язык программирования, который будет использоваться в процессе реализации проекта. Особое внимание будет уделено характеристикам каждой из выбранных технологий, их возможностям и ограничениям, а также тому, как они соответствуют требованиям проекта. В конце раздела будут приведены обоснования выбора данных средств разработки, включая их преимущества, совместимость и удобство использования в контексте поставленных задач. Это позволит понять, почему именно эти инструменты были выбраны для достижения оптимальных результатов в проекте.

## Среда разработки (IDE)

Среда разработки (IDE) играет значимую роль в процессе создания проекта. Удобный интерфейс, встроенные инструменты для отладки и тестирования, а также поддержка расширяемости делают работу более продуктивной. Выбор IDE должен учитывать особенности проекта и предпочтения команды разработчиков.

Средой разработки данной курсовой работы было выбрано Visual Studio Code, имеющий понятный и функциональный интерфейс (рисунок 2.2)

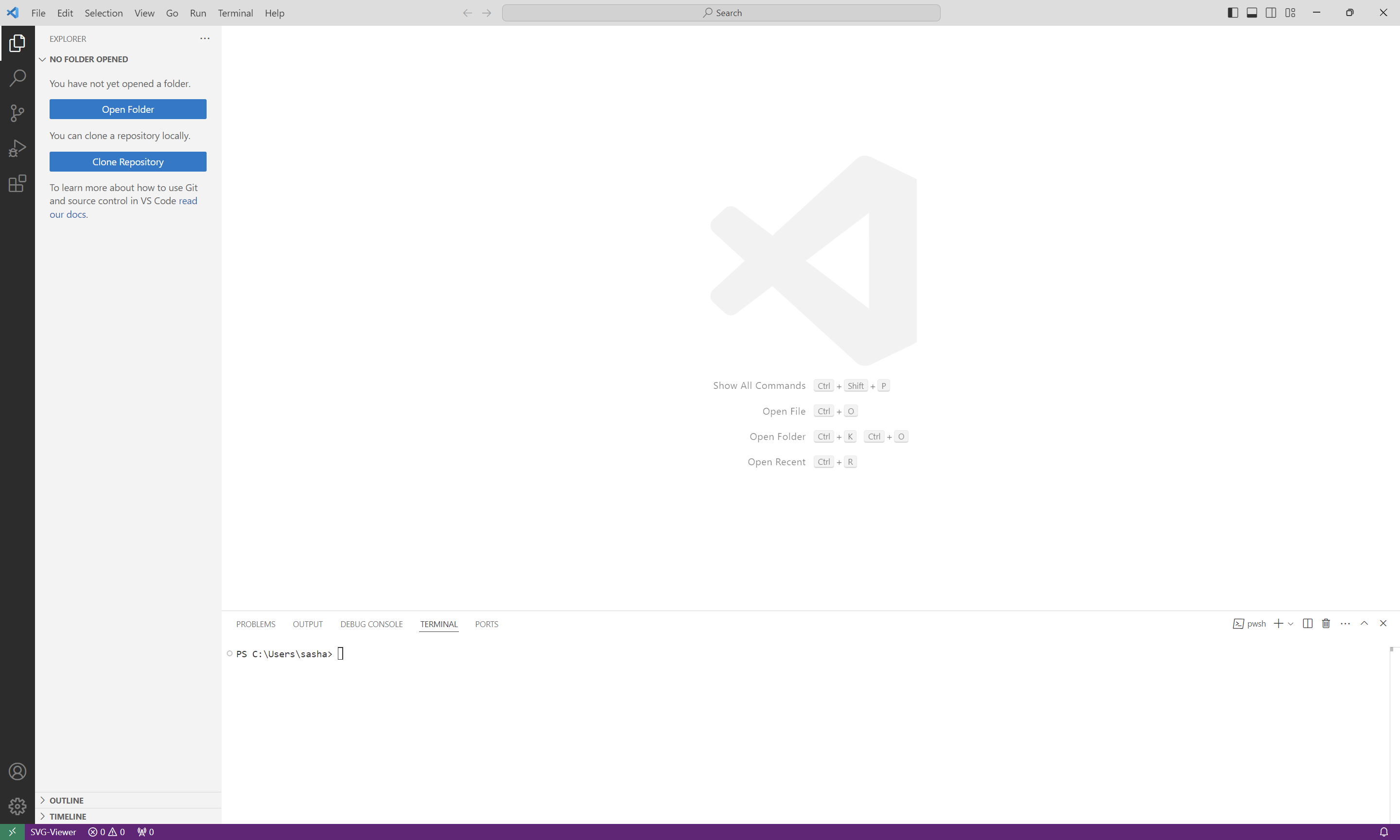


Рисунок 2.2 Интерфейс Visual Studio Code.

Visual Studio Code (VS Code) был выпущен Microsoft в апреле 2015 года. Первоначально позиционировался как лёгкий редактор для веб-разработки, но вскоре благодаря широким возможностям и поддержке множества языков программирования стал универсальным инструментом для разработчиков. В 2016 году проект стал открытым и получил большое сообщество разработчиков.

Первая версия Visual Studio Code, выпущенная в 2015 году, предоставила базовую поддержку веб-технологий, таких как HTML, CSS и JavaScript. В версии 1.x, охватывающей период с 2016 по 2021 год, были добавлены расширения, интеграция с Git, поддержка отладки и улучшения в автодополнении кода. Современные версии, начиная с 2022 года, включают поддержку работы с удалёнными серверами и контейнерами, а также улучшения интерфейса и производительности.

Visual Studio Code — это кроссплатформенный редактор, который работает на Windows, macOS и Linux. Он поддерживает множество языков программирования через плагины, обеспечивая широкую языковую поддержку. Встроенная интеграция с Git позволяет работать с системами контроля версий. Несмотря на свои мощные возможности, VS Code остаётся легким и отзывчивым. Благодаря обширной библиотеке расширений, редактор можно адаптировать под любые нужды разработчика.

## PlatformIO история, версии, достоинства

Чтобы легко запускать написанный код на Arduino было выбрано расширение PlatformIO для Visual Studio Code.

PlatformIO – это открытая экосистема для разработки встроенного программного обеспечения (embedded) с поддержкой множества платформ и микроконтроллеров. Проект был запущен в 2014 году для упрощения разработки программ для микроконтроллеров и быстрого управления зависимостями. Благодаря интеграции с Visual Studio Code и другими IDE, PlatformIO быстро завоевал популярность среди разработчиков встраиваемых систем. Первая версия, выпущенная в 2014 году, предоставила базовые возможности для управления проектами и компиляции кода для нескольких микроконтроллеров. В PlatformIO Core 3.x, выпущенной в 2016 году, была добавлена поддержка библиотек, интеграция с Visual Studio Code и улучшена система сборки. Современные версии 4.x и 5.x расширили поддержку платформ и микроконтроллеров, улучшили отладочные возможности и добавили поддержку удалённой разработки.

PlatformIO поддерживает множество платформ и совместим с более чем 800 микроконтроллерами, включая Arduino, ESP32 и STM32. Интеграция с Visual Studio Code предоставляет мощные инструменты разработки прямо внутри этой популярной среды. Он обеспечивает автоматическое управление зависимостями, библиотеками и сборкой проекта. Также есть поддержка встроенной отладки и мониторинга работы микроконтроллеров. PlatformIO является кроссплатформенным и работает на Windows, macOS и Linux.

## Процесс прошивки Arduino

## Язык программирования

Выбор языка программирования является критически важным этапом, так как он определяет удобство и эффективность разработки. Например, если выбранный язык обладает хорошей поддержкой библиотек и инструментов для работы с необходимыми технологиями, это ускорит процесс реализации проекта и повысит его качество.

На языке программирования Arduino, основанном на C++, удобно разрабатывать проекты для микроконтроллеров благодаря множеству библиотек и удобному синтаксису. Эти библиотеки упрощают работу с различными аппаратными компонентами, такими как датчики, моторы и светодиоды, позволяя легко взаимодействовать с физическим миром.

История C++ началась в 1980-х годах с разработки Бьёрном Страуструпом как расширения языка C, чтобы обеспечить поддержку объектов и классов. Первая версия была представлена в 1985 году, и язык быстро завоевал популярность благодаря сочетанию структурного и объектно-ориентированного программирования.C++98 (1998 г.) – первая стандартизированная версия, в которую вошли основные концепции языка.

– C++03 (2003 г.) – корректировка стандарта C++98.

– C++11 (2011 г.) – важное обновление, добавившее новые функции, такие как лямбда-выражения, auto-типизация и умные указатели.

– C++14 (2014 г.) – улучшения и исправления, основанные на C++11.

– C++17 (2017 г.) – добавлены новые библиотеки и улучшения производительности.

– C++20 (2020 г.) – одно из самых значительных обновлений, включающее концепции, модули и корутины.

Программы для Arduino называются скетчами и сохраняются в файлах с расширением .ino. Эти файлы проходят обработку специальным препроцессором Arduino, который автоматически подключает необходимые библиотеки и преобразует код в подходящий для компиляции формат. Это делает разработку более интуитивной и доступной даже для начинающих программистов.

Скетчи включают функции, специфичные для работы с аппаратными компонентами, что упрощает реализацию проектов в области робототехники, автоматизации и Интернета вещей. Это делает платформу Arduino популярной среди энтузиастов электроники, инженеров и преподавателей, стремящихся обучать основам программирования и электроники.Основные элементы языка программирования Arduino:

1 setup(). Это функция, которая вызывается один раз при запуске микроконтроллера. В ней выполняются начальные настройки и инициализация компонентов, таких как порты ввода-вывода, подключенные устройства и параметры работы. Например, здесь можно установить режимы работы пинов (входные или выходные), инициализировать последовательный порт для обмена данными или настроить начальные значения переменных.

2 loop(). Эта функция вызывается в бесконечном цикле сразу после завершения выполнения функции setup(). В ней размещается основная логика программы, которая выполняется на протяжении всего времени работы микроконтроллера. Здесь можно реализовывать различные действия, такие как считывание данных с датчиков, управление actuators, обработка входных сигналов и выполнение условий. Поскольку функция loop() непрерывно повторяется, она позволяет реагировать на изменения окружающей среды и поддерживать активное взаимодействие с подключенными компонентами. Это делает программу динамичной и способной выполнять задачи в реальном времени.

3 Библиотеки. Arduino предлагает обширный набор библиотек функций для управления различными компонентами, такими как датчики, актуаторы, дисплеи и другие устройства. Эти библиотеки значительно упрощают взаимодействие с аппаратными средствами, позволяя разработчикам сосредоточиться на логике программы, а не на низкоуровневых деталях управления. Кроме стандартных библиотек, предоставляемых Arduino, разработчики имеют возможность создавать свои собственные пользовательские библиотеки. Это позволяет не только повторно использовать код в различных проектах, но и делиться им с другими разработчиками. Пользовательские библиотеки помогают организовать код, делают его более понятным и облегчают процесс разработки, особенно при работе над крупными проектами или в командах.

Использование уже существующего высокоуровневого языка программирования, такого как C++, значительно снижает порог входа для новичков. Это позволяет большему числу людей легко создавать свои проекты на Arduino. Высокоуровневый синтаксис и доступные библиотеки упрощают процесс разработки, позволяя сосредоточиться на концепциях и идеях, а не на сложностях низкоуровневого программирования. Кроме того, наличие обширной документации, примеров кода и активного сообщества способствует быстрому обучению и решению возникающих вопросов. Это делает Arduino популярным инструментом как для начинающих, так и для опытных разработчиков, желающих реализовать свои идеи в сфере электроники, робототехники и Интернета вещей (IoT).

## Обоснование выбора программной платформы

Важным аспектом программной платформы Arduino являются расширенные возможности микроконтроллеров, которые обеспечивают широкий спектр функциональности для разработчиков. Микроконтроллеры Arduino поддерживают различные интерфейсы, такие как I2C, SPI и UART, что позволяет легко интегрироваться с разнообразными периферийными устройствами. Эти интерфейсы обеспечивают высокую скорость передачи данных и позволяют подключать несколько устройств одновременно, что является критически важным при создании сложных встроенных систем.

Используя возможности этих интерфейсов, разработчики могут работать с множеством сенсоров, таких как датчики температуры, влажности, давления и освещения, а также с актуаторами, такими как моторы, реле и светодиоды. Это открывает широкие горизонты для реализации различных проектов – от простых устройств до сложных автоматизированных систем.

Кроме того, Arduino предоставляет обширную библиотеку программного обеспечения, которая упрощает взаимодействие с периферийными устройствами. Это позволяет разработчикам сосредоточиться на логике и функциональности своих проектов, а не на низкоуровневом программировании. Таким образом, Arduino становится идеальной платформой для обучения и прототипирования, позволяя как новичкам, так и опытным инженерам реализовывать свои идеи быстро и эффективно.

С учетом поддержки различных коммуникационных модулей, таких как Wi-Fi и Bluetooth, разработчики могут создавать проекты с удаленным управлением и мониторингом. Это делает Arduino особенно привлекательным для разработки IoT-устройств (интернета вещей), что открывает новые возможности для автоматизации и интеграции в повседневную жизнь.

Таким образом, программная платформа Arduino представляет собой мощный инструмент для разработки встроенных систем и приложений. С ее помощью разработчики могут легко создавать программы для микроконтроллеров Arduino, используя удобную интегрированную среду разработки, стандартный язык программирования и обширную библиотеку функций. Arduino делает процесс программирования доступным и простым, что позволяет как новичкам, так и опытным разработчикам быстро создавать разнообразные встроенные приложения. Также стоит отметить и ее постоянное развитие. Arduino остается в центре инноваций в области встраиваемых систем, активно внедряя новые технологии и поддерживая требования современных разработчиков. Благодаря открытой архитектуре и плодотворному взаимодействию с сообществом, платформа продолжает привлекать внимание как опытных инженеров, так и стартапов, исследующих новые горизонты в области встраиваемых технологий.

# 3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА

Процесс лазерной резки требует высокой точности и безопасности. При работе с лазерными системами необходимо учитывать множество факторов, чтобы избежать нежелательных последствий, таких как повреждение оборудования или материалов. Одной из ключевых проблем, с которой сталкиваются операторы лазерных резаков, является контроль за горением заготовки. Это явление может привести к неконтролируемому распространению пламени, что, в свою очередь, может вызвать серьезные повреждения оборудования и даже травмы операторов.

В связи с этим разработка компактного и надежного устройства для определения горения заготовки в процессе лазерной резки является крайне актуальной задачей. Такое устройство может стать важным инструментом в обеспечении безопасности и эффективности процесса лазерной резки. Оно способствует снижению числа неисправностей и повышению общей производительности. Применение передовых технологий, таких как инфракрасные датчики пламени, позволяет быстро и точно определять наличие горения. Эти датчики способны реагировать на изменения температуры и светового излучения, характерные для пламени, что обеспечивает мгновенную реакцию на инциденты.

Компактные и недорогие датчики могут быть интегрированы в лазерные резаки или использоваться как портативные устройства. Это делает их доступными для широкой аудитории, включая малые и средние предприятия, которые часто сталкиваются с необходимостью оптимизации производственных процессов без значительных затрат. Внедрение таких устройств может также способствовать улучшению условий труда, снижая уровень стресса и повышая уверенность операторов в безопасности оборудования.

В итоге, создание эффективного и доступного устройства для определения горения заготовки может существенно снизить риски, связанные с лазерной резкой. Это не только повышает безопасность всех участников процесса, но и способствует улучшению качества продукции, снижению производственных простоев и затрат на ремонт оборудования. Таким образом, внедрение подобных технологий является важным шагом на пути к более устойчивым и безопасным производственным процессам.

## Углубленный анализ инфракрасных датчиков пламени

Инфракрасные датчики пламени разработаны специально для обнаружения пламени в ближнем инфракрасном диапазоне (700-1100 нм), что делает их идеальными инструментами для контроля за горением в процессе лазерной резки. Эти датчики способны быстро и точно определять наличие пламени, что критически важно для обеспечения безопасности процесса.

Однако стоит отметить, что одной из проблем датчиков является их чувствительность к шумам и различным внешним факторам. Для повышения точности измерений рекомендуется проводить многократное снятие показаний. Полученные данные следует обрабатывать с использованием соответствующих алгоритмов, и только после этого выводить результаты на индикаторные светодиоды. Такой подход поможет снизить вероятность ошибок и повысить надежность показаний.

Для обеспечения устойчивости к возможным сбоям в работе системы рекомендуется использовать несколько инфракрасных датчиков одновременно. В случае выхода одного из них из строя или его отключения от платы Arduino, обработка данных может основываться на показаниях других датчиков. Это гарантирует, что даже при неисправности одного из устройств, система продолжит функционировать и предоставлять актуальные результаты.

## 3.2 Калибровка датчиков

Калибровка датчиков инфракрасного пламени — важный этап для обеспечения точности и надежности устройства. Рассмотрим основные шаги калибровки, используя приведенный код. Сначала необходимо настроить окружение, используя контролируемые условия горения для измерения эталонных значений. Это поможет установить базовые уровни для датчиков. Затем нужно считывать значения с каждого датчика и усреднять их, чтобы определить базовый уровень для неактивного состояния. В коде это реализовано через массивы, которые помогают вычислять средние значения.

Далее, следует определить пороги. Используйте увеличенный порог для определения скачков, которые могут свидетельствовать о наличии пламени, что поможет избежать ложных срабатываний. Затем важно обрабатывать данные, постоянно обновляя текущие показания и сравнивая их с базовыми значениями. Отклонения определяются с помощью специальных переменных.

Регулярная проверка калибровки необходима для поддержания точности устройства, учитывая возможные изменения характеристик датчиков со временем. На основе результатов калибровки следует корректировать значения, такие как максимальное и минимальное, а также чувствительность, для оптимальной работы устройства. Эти шаги обеспечат высокую точность и надежность устройства для обнаружения пламени.

## 3.3 Пользовательский интерфейс

Создание простого и интуитивно понятного пользовательского интерфейса является ключевым аспектом успешного проекта. Четкое и понятное отображение результатов на индикаторных светодиодах существенно повысит удобство использования устройства, позволяя пользователям легко воспринимать необходимую информацию.

Кроме того, на пользовательский интерфейс можно выводить данные о различных ошибках, возникающих в процессе работы. Это не только упростит процесс диагностики и устранения неполадок, но и позволит пользователям быстрее реагировать на возможные проблемы. В результате они смогут оперативно принимать меры для устранения неисправностей, что значительно повысит надежность устройства. Также стоит отметить, что наличие информативных сообщений об ошибках и состоянии системы может снизить вероятность неправильного использования устройства, обеспечивая тем самым более безопасную эксплуатацию. Поэтому продуманный интерфейс не только улучшает взаимодействие с пользователем, но и способствует повышению общей эффективности и долговечности устройства.

Таким образом, такие устройства могут стать важным элементом в рамках комплексного подхода к безопасности в промышленности. Их интеграция в программы профилактики неисправностей может включать установку датчиков на производственных предприятиях и в лабораториях, где возможно возникновение горения. Это создает дополнительные возможности для повышения общественной безопасности и формирования культуры ответственного отношения к процессу лазерной резки.

# 4 ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ПРОГРАММЫ

## 4.1 Обоснование и описание функциональных возможностей

**4.1.1** Запуск и инициализация

**Функция:** Запуск и инициализация системы после подачи питания или перезагрузки.

**Обоснование:** В момент включения алкотестера требуется выполнить ряд начальных настроек и подготовительных действий, таких как настройка портов на ввод или вывод и задание калибровочных параметров.

**Реализация:** При включении устройства микроконтроллер Arduino будет инициализировать все подключенные компоненты, устанавливать начальные значения и запускать цикл обработки ввода.

**4.1.2** Измерение уровня излучения

**Функция:** Считывание данных с инфракрасных датчиков пламени для определения признаков наличия горения.

**Обоснование:** Это ключевая функция системы, так как именно здесь происходит сбор данных, необходимых для определения наличия горения. Инфракрасные датчики пламени осуществляют измерения, и полученные данные передаются в программное обеспечение для дальнейшей обработки.

**Реализация:** Программное обеспечение будет периодически опрашивать инфракрасные датчики пламени, считывать значения и преобразовывать их в интенсивность пламени с использованием заранее определенных коэффициентов.

**4.1.3** Анализ результатов

**Функция:** Анализ данных для определения характера скачков на датчиках, чтобы определить, это шум или горение.

**Обоснование:** Анализ характера скачков на датчиках позволяет отличить шум от реального горения. Это важно для предотвращения ложных срабатываний и повышения точности системы.

**Реализация:** Программное обеспечение будет анализировать данные с инфракрасных датчиков пламени, чтобы определить характер скачков. Если скачки имеют характер шума, система будет их игнорировать. Если скачки указывают на наличие горения, система будет принимать.

**4.1.4** Отображение результатов

**Функция:** Индикация обнаружения пламени.

**Обоснование:** Пользователь должен видеть результаты измерения и рекомендации для принятия соответствующих решений. Отображение информации на индикаторных светодиодах делает использование системы удобным и информативным.

**Реализация:** При измерении если прибор обнаружит признаки наличия горения, результат выведется на индикаторный светодиод.

**4.1.5** Отображение ошибок

**Функция:** Вывод на экран возникающих в процессе работы ошибок.

**Обоснование:** Отображение ошибок способно облегчить ремонт аппаратных компонентов устройства, а также предупредить пользователя о том, что данные могут быть некорректными.

**Реализация:** При обнаружении ошибки, загорится светодиод, предупреждающий о наличии ошибки.

Таким образом, проектирование функциональных возможностей программного обеспечения для алкотестера на базе Arduino с использованием датчиков паров спирта MQ-3 представляет собой многоэтапный процесс, который охватывает различные аспекты разработки. В первую очередь, необходимо создать и реализовать функции, которые обеспечивают точное измерение и анализ уровня алкоголя в выдыхаемом воздухе. Это включает в себя не только обработку данных, получаемых от датчика, но и их корректную интерпретацию с учетом возможных погрешностей и внешних факторов. Кроме того, проект предполагает внедрение механизмов, которые будут предоставлять пользователю рекомендации на основе полученных результатов. Например, устройство сможет выдавать советы о том, стоит ли садиться за руль, или предлагать альтернативные способы передвижения в случае превышения допустимого уровня алкоголя. Также важной частью функционала станет система оповещения, которая будет информировать пользователя о состоянии его алкоголя в крови. Это может быть реализовано через звуковые сигналы, вибрацию или визуальные уведомления на экране, что делает устройство более интуитивно понятным и доступным для использования в стрессовых ситуациях. В результате, разработанные функции позволят создать полноценное устройство для контроля уровня алкоголя, которое не только обеспечит безопасность на дорогах, но и будет способствовать общей безопасности в обществе, снижая риски, связанные с вождением в нетрезвом виде.

# 5 АРХИТЕКТУРА РАЗРАБАТЫВАЕМОЙ ПРОГРАММЫ

Для курсового проекта была использована оригинальная Arduino UNO, которая обеспечивает все необходимые функции. Этот выбор обусловлен её доступностью и простотой работы, что идеально подходит для выполнения поставленных задач. Компоненты проекта соединяются с помощью макетной платы, что упрощает настройку и модификацию схемы, особенно на этапе разработки и тестирования. Такой подход обеспечивает гибкость и упрощает эксперименты с различными конфигурациями устройства.

Макетная плата, или breadboard – это беспаечная плата для монтажа. Универсальный инструмент для моделирования прототипов устройств. Альтернатива создания схем без применения паяльника. Монтажную плату применяют для конструирования, отладки и тестирования будущей схемы устройства при разных условиях подключения и эксплуатации. Внешний вид макетной платы представлен на рисунке 5.1.

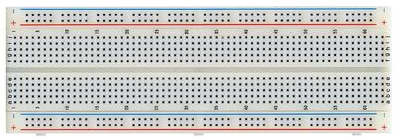


Рисунок 5.1 – Макетная плата

Макетная плата состоит из пластиковой пластины-основания с большим количеством отверстий. Расстояние между отверстиями – 2,54 мм, их диаметр – 0,8 мм. Внутри расположены два вида дорожек: контактные группы пластин и вертикальные дорожки для подачи питания. Ряды металлических пластин-рельс имеют по пять клипс каждая. Отверстия, в которые будут вставляться ножки элементов, соединяют клипсы и таким образом замыкают схему. Проводник, подключенный к отверстию в одном из рядов, одновременно соединяется с остальными контактами этого ряда. На одной рельсе можно подключить до 5 элементов, которые будут связаны между собой. Две рельсы одного ряда изолированы друг от друга. Рельсы питания расположены вертикально по краям. Как правило, синий цвет линии указывает, что нужно подавать «-» напряжения, а красный – «+». Чтобы легче работать с макетной платой, на нее производители наносят обозначения дорожек латинскими буквами и цифрами. Так проще описывать процесс сборки или писать инструкции для подключений. На рисунке 5.2 представлено обозначение дорожек на макетной плате.

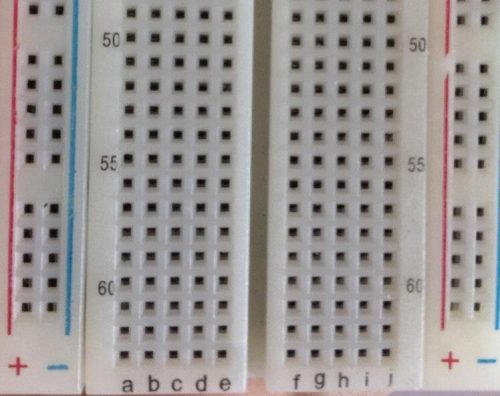


Рисунок 5.2 – Обозначение дорожек на макетной плате

Макетная плата – это современный вариант сборки схем, который позволяет легко изменять варианты схемы, добавлять компоненты и проверять теории и способы подсоединения. Если вы паяете и что-то неправильно соединили или решили что-то изменить в схеме – придется перепаивать. На breadboard процесс реконструкции, отладки и модификации осуществляется мгновенно: снял, подключил по-другому. Однако беспаечный способ соединения элементов схемы не такой надежный, как при сцеплении припоем. Например, при постоянной вибрации устройства контакты будут понемногу ослабляться. Кроме того, внешний вид с висящими проводами требует большего корпуса для проекта.

## 5.1 Подключение устройств к Arduino

Так как прямое подключение устройств к Arduino невозможно, для этих целей будут использоваться провода "папа-папа", которые относится к типам разъемов, используемым для подключения проводов и периферийных устройств к плате Arduino или другим микроконтроллерам. Эти разъемы могут быть представлены в виде "мама" (female) и "папа" (male), где "мама" представляет собой разъем с отверстиями для вставки, а "папа" – выступающий разъем для вставки в "маму".

Для подключения пяти инфракрасных датчиков к Arduino, предположим, что у вас есть пять датчиков, каждый из которых имеет три вывода: GND, VCC и AO (аналоговый выход). Подключение датчиков смотр на рисунке 5.3:

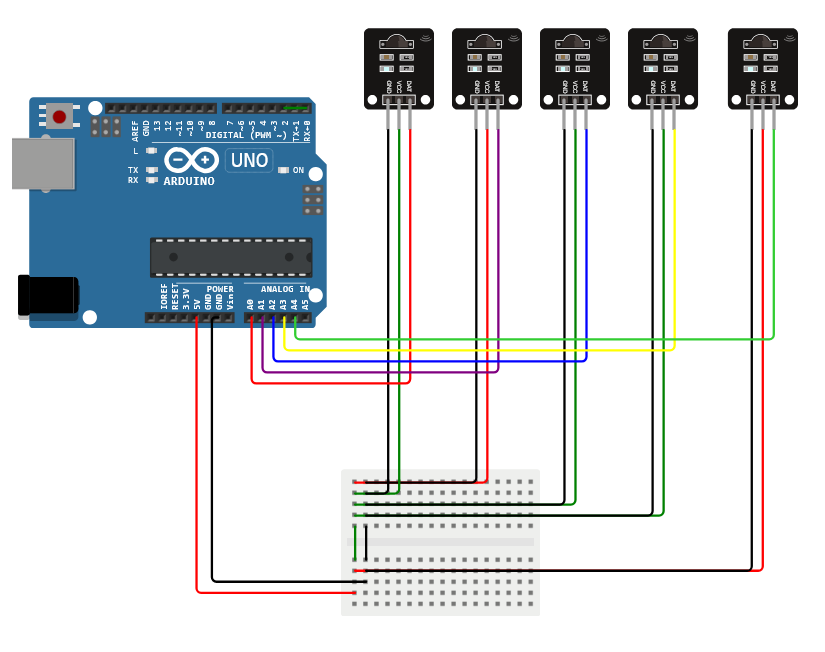


Рисунок 5.3 – Подключение всех датчиков к плате

Подключение питания:

– Все выводы GND датчиков подключаются к общему GND на Arduino.

– Все выводы VCC датчиков подключаются к общему 5V или 3.3V на Arduino (в зависимости от требований датчиков).

Подключение аналоговых выходов:

– Разъем AO первого датчика подключается к аналоговому порту A0 на Arduino.

– Разъем AO второго датчика подключается к аналоговому порту A1.

– Разъем AO третьего датчика подключается к аналоговому порту A2.

– Разъем AO четвертого датчика подключается к аналоговому порту A3.

– Разъем AO пятого датчика подключается к аналоговому порту A4.

Для подключения светодиода к Arduino понадобится резистор (обычно 220 Ом или 330 Ом), схема подключения указана ниже (Рисунок 5.4).

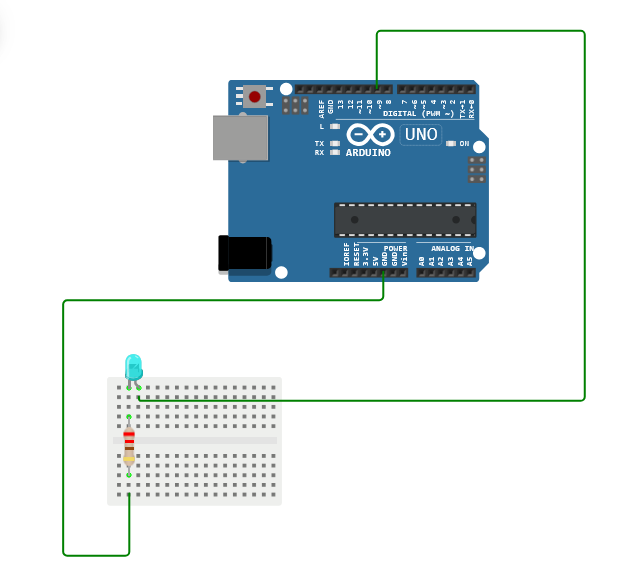


Рисунок 5.4 – Подключение информационного светодиода

На Arduino UNO имеется всего два пина 5V, что может ограничивать количество подключаемых устройств, особенно если требуется питание для нескольких компонентов. В этой ситуации использование макетной платы становится особенно актуальным и удобным. Макетная плата, или прототипная плата, позволяет эффективно организовать подключение множества различных элементов, таких как датчики, светодиоды и модули, обеспечивая при этом простоту и гибкость в настройке схемы. Одним из ключевых преимуществ макетной платы является возможность быстрого монтажа и демонтажа компонентов. Это делает процесс разработки более динамичным, так как позволяет легко вносить изменения в соединения без необходимости пайки, что экономит время и усилия. Например, если нужно протестировать новый датчик или изменить конфигурацию, можно сделать это, просто переставив провода. Кроме того, макетная плата способствует лучшему управлению пространством и уменьшает беспорядок на рабочем месте. Все соединения можно организовать аккуратно и логично, что облегчает отладку и поиск возможных ошибок в схеме. Также такая платформа подходит как для новичков, так и для опытных разработчиков, предоставляя возможность экспериментировать с различными концепциями и проектами без значительных затрат.

## 5.2 Общая структура программы

Архитектура программного обеспечения разрабатываемого алкотестера имеет модульную структуру, что обеспечивает организацию программы в виде набора отдельных функциональных блоков, выполняющих определенные задачи и взаимодействующих с другими блоками для обеспечения корректной работы устройства. Ниже описаны основные модули программы:

1 модуль инициализации и настройки отвечает за первоначальную инициализацию платы Arduino, настройку портов ввода-вывода и инициализацию подключенный устройств;

2 модуль измерения уровня алкоголя отвечает за считывание данных с конкретного датчика паров спирта MQ-3;

3 модуль анализа данных отвечает за обработку данных, полученных от датчиков для дальнейшего вывода этих данных на дисплей;

4 модуль вывода данных отвечает за отображение пользовательского интерфейса и вывод обработанных данных, полученных с датчиков, на экран;

5 модуль обработки ошибок отвечает за обработку данных, поступающих от датчиков, и определение ошибок в работе сенсоров на основе этих данных. В случае обнаружения ошибки, ей присваивается определенный код, который далее выводится на экран;

6 модуль вывода ошибок отвечает за отображение кодов ошибок на дисплее;

7 основной цикл программы отвечает за работу устройства и взаимодействие остальных модулей программы между собой.

Такая организация программы упрощает ее разработку и поддержку, а также упрощает поиск и исправление ошибок в ее работе.

## 5.3 Описание функциональной схемы программы

Функциональная схема – это графическая (приложение Б) или текстовое представление взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Она используется для визуализации и анализа функциональных аспектов системы, позволяя легче понять, как система выполняет определенные задачи или функции. Ниже представлено текстовое описание функциональной схемы программы.

**5.3.1** Начало программы:

– запуск программы на микроконтроллере Arduino;

– настройка портов микроконтроллера и инициализация подключенных устройств;

– инициализация последовательного порта для взаимодействия с компьютером, подключенным через USB.

**5.3.2** Основной цикл:

– программная задержка для уменьшения количества выводимых сообщений;

– проверка подключенных датчиков на наличие ошибок в подключении;

– считывание данных с датчиков, для каждого из датчиков считывается по 20 значений, а затем находится их среднее арифметическое. Данное действие необходимо для корректировки значений и получения более точных данных;

– проверка значений, нахождение аномалий в сигнале;

– возврат к началу основного цикла.

## 5.4 Описание блок-схемы алгоритма программы

Блок-схема – это схематичное представление процесса (приложение В), системы или компьютерного алгоритма. Блок-схемы часто применяются в разных сферах деятельности, чтобы документировать, изучать, планировать, совершенствовать и объяснять сложные процессы с помощью простых логичных диаграмм. Для построения блок-схем применяются прямоугольники, овалы, ромбы и некоторые другие фигуры (для обозначения конкретных операций), а также соединительные стрелки, которые указывают последовательность шагов или направление процесса. Ниже представлено текстовое описание блок-схемы алгоритма программы.

**5.4.1** Подключение библиотек, инициализация устройств и глобальных переменных программы:

– подключение библиотеки Arduino;

– создание глобальных переменных для хранения данных с датчиков, состояния светодиодов и пороговых значений;

– создание массивов для хранения чтений с датчиков, индексов, сумм, средних значений, базовых линий и флагов обнаружения скачков.

**5.4.2** Установка программы (setup):

– инициализация последовательного порта и установка скорости его работы;

– инициализация массивов для хранения данных с датчиков и усреднения значений;

– первоначальное считывание данных с датчиков и усреднение значений для установки базовых линий.

**5.4.3** Обработка ошибок в работе датчиков:

– чтение данных с датчиков;

– проверка наличия питания на датчиках (в случае его отсутствия на аналоговом выходе, к которому подключен датчик, будет значение 0). В случае обнаружения ошибки вывод сообщения об ошибке;

– проверка подключения датчиков к аналоговым выходам (в случае отсутствия подключения к устройству на пине будет максимальное возможное для него значение, а именно 1024). В случае обнаружения ошибки вывод сообщения об ошибке;

– выход из функции обработки ошибок.

**5.4.4** Вывод кода ошибки на экран:

– проверка на наличие кода ошибки в переменной err\_code;

– в случае, если ошибок нет, возврат из функции вывода кода ошибки;

– установка размера текста;

– установка курсора в необходимую позицию;

– вывод кода ошибки на экран.

**5.4.5** Чтение данных с датчиков:

– считывание данных с каждого датчика и обновление массивов чтений;

– нахождение среднего арифметического из считанных значений;

– обновление базовых линий с использованием плавной адаптации;

– результата из функции.

**5.4.6** Обнаружение скачков:

– обнаружение скачков в данных с датчиков и установка флага скачка, если разница между текущим значением и базовой линией превышает порог;

– сброс флага скачка, если сигнал вернулся в норму;

– обнаружение засветки датчиков и установка флага засветки, если значение превышает максимальное значение.

**5.4.7** Обработка засветки и сигнализации:

– если засветка обнаружена, включение светодиода через 5 секунд;

– сигнализация при множественных скачках: если количество скачков превышает порог, включение светодиода сигнализации;

– понижение уровня опасности: плавное снижение счётчика скачков через определенное количество циклов.

**5.4.8** Программная задержка:

– программная задержка для стабилизации ввода-вывода;

– возврат к началу основного цикла.

Таким образом, разработанное устройство позволяет определять аномалии во входящих сигналах, такие как резкие скачки и падения, для определения наличия признаков горения. Программа обладает модульной структурой, что облегчает ее поддержку. Также данное устройство способно обрабатывать различные ошибки в работе датчиков, что облегчает его ремонт и эксплуатацию в случае выхода какого-либо датчика из строя.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Стандарт предприятия. Дипломные проекты (работы). Общие требования [Электронный ресурс]. – Режим доступа: https://www.bsuir.by/m/12\_100229\_1\_185586.pdf. – Дата доступа: 23.09.2024

[2] Микропроцессоры и микропроцессорные комплекты интегральных микросхем : справочник. В 2 т. / под ред. В. А. Шахнова. – М. : Радио и связь, 1988. – Т. 1. – 368 с.