Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Архитектура вычислительных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

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

БГУИР КП 1-40 04 01

Студент: Сенько Н. С.

гр. 253505

Руководитель: Калиновская А. А.

Минск 2024

**СОДЕРЖАНИЕ**

[Введение 5](#_Toc178697841)

[1 Архитектура вычислительной системы 6](#_Toc178697842)

[2 Платформа программного обеспечения 12](#_Toc178697843)

[2.1 Язык программирования C++ история, версии, достоинства 12](#_Toc178697844)

[2.2 Visual Studio Code история, версии, достоинства 13](#_Toc178697845)

[2.3 PlatformIO история, версии, достоинства 14](#_Toc178697846)

[​](#_Toc178697847)

# ВВЕДЕНИЕ

Курсовая работа посвящена разработке системы определения горения заготовки в процессе лазерной резки на базе микроконтроллера ESP32. Основная задача — подобрать и проанализировать данные с газовых и инфракрасных датчиков 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 надёжности и точности измерений в реальном времени, что crucial для оптимизации процесса лазерной резки. В-третьих, реализация удалённого 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 через Telegram-бот. Использование Telegram-бота обеспечивает удобный и современный способ взаимодействия с системой, позволяя операторам получать уведомления и управлять процессом с мобильных устройств.

Перечень планируемых задач:

– Подобрать и настроить датчики для фиксации газа и инфракрасного излучения.

– Интегрироваться с интерфейсами передачи данных.

– Создать интерфейсы управления и настройки системы через Telegram-бот.

– Протестировать и отладить систему.

Детальная постановка задачи:

– Изучить техническую документацию используемых датчиков и их интерфейсов.

– Исследовать функциональные возможности ESP32 для выполнения задачи.

– Разработать взаимодействие с системой через Telegram-бот.

– 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.

Тестирование системы должно включать проверку точности анализа в различных условиях, имитирующих реальный процесс лазерной резки. Это может потребовать создания специальных стендов или использования существующего оборудования в тестовом режиме. Проверка корректности передачи данных и работы системы управления через Telegram-бот должна охватывать различные сценарии использования, включая нестандартные ситуации и попытки некорректного ввода данных пользователем.

# 1 АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

* 1. **Структура и архитектура вычислительной системы**
     1. Создание системы сигнализации о горении заготовки на базе ESP32 включает в себя несколько ключевых компонентов и технологий. Основной задачей является своевременное обнаружение признаков горения, что позволяет предотвратить возможные аварии и повысить уровень безопасности на производстве. Для этого будут использоваться датчики серии MQ, которые анализируют состав газов в окружающей среде.

Микроконтроллер ESP32 служит центральным элементом системы, обрабатывая данные, полученные с датчиков. Благодаря встроенным модулям Wi-Fi и Bluetooth, ESP32 обеспечивает удаленный мониторинг и управление системой. Датчик MQ135 отвечает за определение уровня аммиака, сероводорода и бензола. MQ4 специализируется на обнаружении метана и природного газа, в то время как MQ7 чувствителен к угарному газу, что делает его идеальным для выявления CO, выделяющегося при горении.

MQ2 — это популярный газовый датчик, который может обнаруживать различные горючие газы, такие как пропан, бутан, метан и пары спиртов. Он работает на полупроводниковой технологии, изменяя сопротивление при контакте с газами, что делает его полезным для систем обнаружения утечек газа и мониторинга качества воздуха. Планируется провести тесты на реакцию различных датчиков на CO2, включая MQ2, MQ135, MQ4 и MQ7, чтобы определить их чувствительность и устойчивость к изменениям уровня CO2, что поможет улучшить точность и надежность систем мониторинга.

Вся система интегрирована в единую сеть, где данные с датчиков постоянно анализируются. При обнаружении концентрации газов выше установленного порога, ESP32 активирует сигнализацию и отправляет уведомление ответственным лицам. Это позволяет оперативно реагировать на потенциальные угрозы и предотвращать инциденты. Благодаря использованию современных технологий и надежных компонентов, система сигнализации обеспечивает высокий уровень точности и надежности.

* + 1. Создание системы сигнализации о горении заготовки на базе ESP32 также включает использование инфракрасных технологий для более точного обнаружения огня. Инфракрасный датчик движения HC-SR501 помогает в выявлении тепловых изменений в окружающей среде, что служит дополнительным средством диагностики. Инфракрасный датчик способен обнаруживать инфракрасное излучение, характерное для огня. Это позволяет системе не только реагировать на изменения в газовом составе, но и фиксировать непосредственные признаки горения. Когда датчик HC-SR501 обнаруживает движение или тепловое излучение, система получает дополнительный сигнал для активации сигнализации.

Интеграция инфракрасного обнаружения в систему повышает ее надежность. Использование как газовых, так и инфракрасных датчиков обеспечивает всесторонний подход к мониторингу состояния заготовки, позволяя более эффективно предотвращать аварийные ситуации.

* 1. **ESP32 история, версии и достоинства**
     1. Микроконтроллеры ESP32 – это энергоэффективные системы на кристалле, разработанные китайской компанией Espressif Systems. В настоящее время выпускаются в основном пять линеек продукции: ESP8266, ESP32, ESP32-S\*, ESP32-C\* и ESP32-H. Серии ESP32 (изображен на рисунке 1.1) и ESP32-S\* основаны на процессорных ядрах с архитектурой компании Tensilica, когда в ESP32-C\* и ESP32-H используются ядра с открытой архитектурой RISC-V. Более глубоко рассмотрим линейку ESP32, микроконтроллер которой будет использоваться при выполнении курсового проекта.



Рисунок 1.1 – Микроконтроллер ESP32.

ESP32 представляют собой систему на кристалле (SoC), которая была выпущена после ESP8266 в 2016 году с интегрированными контроллерами радиосвязи Wi-Fi, Bluetooth и Thread. По сравнению с предшествующим микроконтроллером в ESP32 производитель выполнил модуль при помощи технологии 40 нм, что позволило на одной плате поместить больше цифровых узлов, а также позволило снизить энергопотребление, сделав работу устройств с батарейным или аккумуляторным питанием более длительной.

* + 1. Основные характеристики микроконтроллера ESP32: Процессор: Двухъядерный 32-битный процессор Xtensa LX6 с тактовой частотой до 240 МГц. Память: 520 КБ SRAM, 448 КБ ROM, до 16 МБ внешней Flash-памяти. Беспроводные интерфейсы: Wi-Fi 802.11 b/g/n (2.4 ГГц), Bluetooth 4.2 и Bluetooth Low Energy (BLE). Периферийные интерфейсы: GPIO (до 34 программируемых выводов), SPI, I2C, I2S, UART, CAN 2.0, Ethernet MAC, SDIO/MMC host контроллер. Аналоговые интерфейсы: 12-битный АЦП (до 18 каналов), 8-битный ЦАП (2 канала), датчик температуры, сенсорные емкостные входы (до 10 каналов). Безопасность: Аппаратное ускорение шифрования (AES, SHA, RSA, ECC), генератор случайных чисел. Режимы энергосбережения: Active mode, Modem-sleep mode, Light-sleep mode, Deep-sleep mode.

Микроконтроллер ESP32 нашел широкое применение в различных областях, таких как: Интернет вещей (IoT): благодаря встроенным Wi-Fi и Bluetooth модулям, ESP32 идеально подходит для создания умных устройств, способных подключаться к сети и взаимодействовать друг с другом. Носимая электроника: низкое энергопотребление и компактные размеры делают ESP32 отличным выбором для создания фитнес-трекеров, умных часов и других носимых устройств. Домашняя автоматизация: ESP32 часто используется в системах умного дома для управления освещением, климатом, безопасностью и другими параметрами. Промышленная автоматизация: благодаря широкому набору интерфейсов и высокой производительности, ESP32 может применяться в системах мониторинга и управления промышленным оборудованием. Робототехника: мощный процессор и встроенные беспроводные интерфейсы позволяют использовать ESP32 в качестве управляющего модуля для роботов и дронов.

* + 1. Программирование ESP32 может осуществляться с использованием различных средств разработки и языков программирования. Наиболее популярные варианты включают: Arduino IDE: благодаря наличию библиотек и простому интерфейсу, Arduino IDE позволяет быстро начать разработку для ESP32 даже начинающим программистам. Espressif IoT Development Framework (ESP-IDF): официальная среда разработки от производителя, предоставляющая наиболее полный доступ к возможностям микроконтроллера. MicroPython и CircuitPython: интерпретируемые языки программирования, позволяющие писать код для ESP32 на Python-подобном языке. Платформа PlatformIO: универсальная среда разработки, поддерживающая множество плат и фреймворков, включая ESP32.

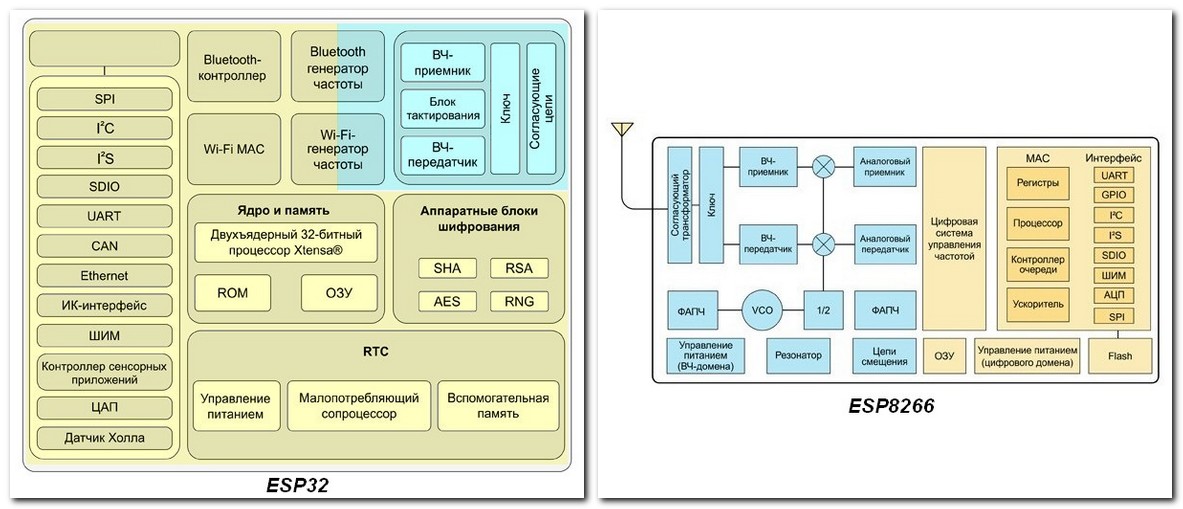


Рисунок 1.2 – Микроархитектура контроллера ESP32.

* + 1. Микроконтроллеры ESP32 – это энергоэффективные системы на кристалле, разработанные китайской компанией Espressif Systems. В настоящее время выпускаются в основном пять линеек продукции: ESP8266, ESP32, ESP32-S\*, ESP32-C\* и ESP32-H. Серии ESP32 (изображен на рисунке 1.1) и ESP32-S\* основаны на процессорных ядрах с архитектурой компании Tensilica, когда в ESP32-C\* и ESP32-H используются ядра с открытой архитектурой RISC-V. Более глубоко рассмотрим линейку ESP32, микроконтроллер которой будет использоваться при выполнении курсового проекта. [1]

Заметим, что на рисунке 1.2 изображены две различные области микроконтроллера: высокочастотная (отмечена голубым цветом) и цифровая части.   
Рассмотрев цифровую составляющую, можем выделить её основные компоненты:

* Центральный процессор (двухъядерный процессор Xtensa 32­bit LX6);
* Встроенная память (FLASH и SRAM);
* Контроллеры беспроводных интерфейсов (Wi-Fi 2.4 МГц и Bluetooth 4.2);
* Цифровая периферия (UART, Ethernet, SPI, I2C и т.д.);
* Модуль RTC, предназначенный для работы камня в режимах с низким энергопотреблением, общего управления питанием, а также включающий в себя всю аналоговую периферию (АЦП, ЦАП, драйвер сенсорных датчиков, осциллятор на 32кГц);
* Аппаратные блоки шифрования данных.

В целом, микроконтроллер ESP32 представляет собой мощное и универсальное решение для широкого спектра задач, связанных с обработкой данных, беспроводной связью и управлением периферийными устройствами. Его высокая производительность, низкое энергопотребление и богатый набор интерфейсов делают его привлекательным выбором для разработчиков встраиваемых систем и IoT-устройств.

* 1. **Датчики MQ серии, преимущества, принципы работы**
     1. Датчики серии MQ работают на основе полупроводниковой технологии, использующей окись олова (SnO2) в качестве чувствительного элемента. В нормальных условиях это вещество имеет высокое сопротивление, но при контакте с определёнными газами сопротивление снижается.

Каждый датчик оснащён нагревателем, который поддерживает необходимую температуру для реакции. При попадании газа на чувствительный элемент изменяется проводимость, и это изменение можно измерить и интерпретировать как концентрацию газа. Разные модели MQ имеют разные сенсоры, оптимизированные для обнаружения конкретных газов, таких как метан, аммиак или угарный газ.

* + 1. Датчики серии MQ имеют свои преимущества и недостатки. Преимущества: они доступны и относительно недорогие, могут обнаруживать различные газы, просты в использовании и подходят для систем мониторинга воздуха. Недостатки: низкая избирательность, могут не точно различать газы, долгое время отклика, высокое энергопотребление из-за нагревательного элемента и необходимость регулярной калибровки.
  1. **Определение лучшего варианта для детекции CO**
     1. В ходе проведения тестирования датчиков серии MQ для определения их эффективности в обнаружении угарного газа (CO) были использованы модели MQ2, MQ135, MQ4 и MQ7. Перед проведением испытаний необходимо было прогреть датчики в течение 24 часов. Испытания проводились на Arduino Uno с лабораторным источником питания. Каждый датчик был подвергнут одинаковым условиям, где уровень угарного газа постепенно увеличивался, а показания записывались для анализа.

Результаты тестирования представлены в виде графиков, которые наглядно демонстрируют чувствительность каждого датчика к CO. Ожидается, что MQ7 покажет наилучшую реакцию на угарный газ, так как он специально разработан для его обнаружения. Эти данные помогут определить, что MQ7 является оптимальным выбором для систем, требующих точного обнаружения CO.

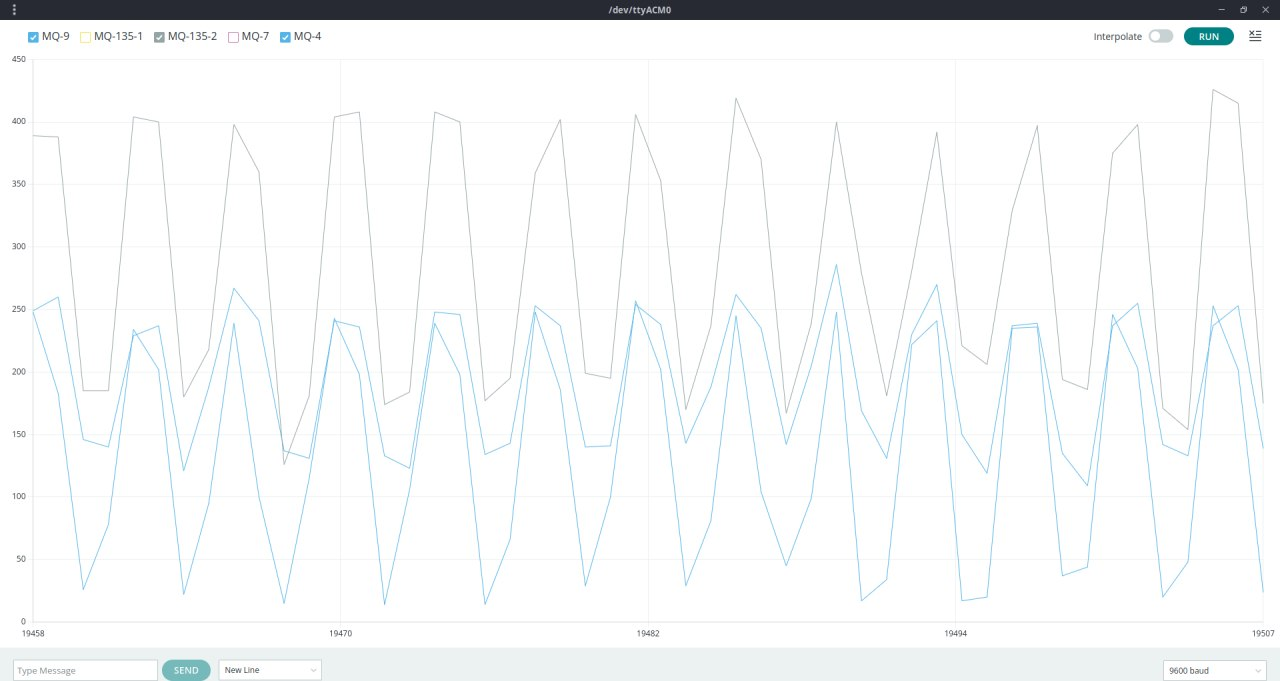


Рисунок 1.3 – Графики реакции датчиков на CO.

Эти испытания позволили выявить сильные и слабые стороны каждого датчика в условиях повышенной концентрации угарного газа. На графиках можно увидеть, что MQ7 демонстрирует наиболее стабильные и точные показания, что подтверждает его пригодность для использования в системах безопасности и контроля качества воздуха. В то время как другие датчики также реагировали на изменения уровня CO, их чувствительность и избирательность оказались ниже. Таким образом, для проектов, где критически важна точность в обнаружении угарного газа, рекомендуется использовать именно MQ7.

* 1. **Конструирование прототипа системы**

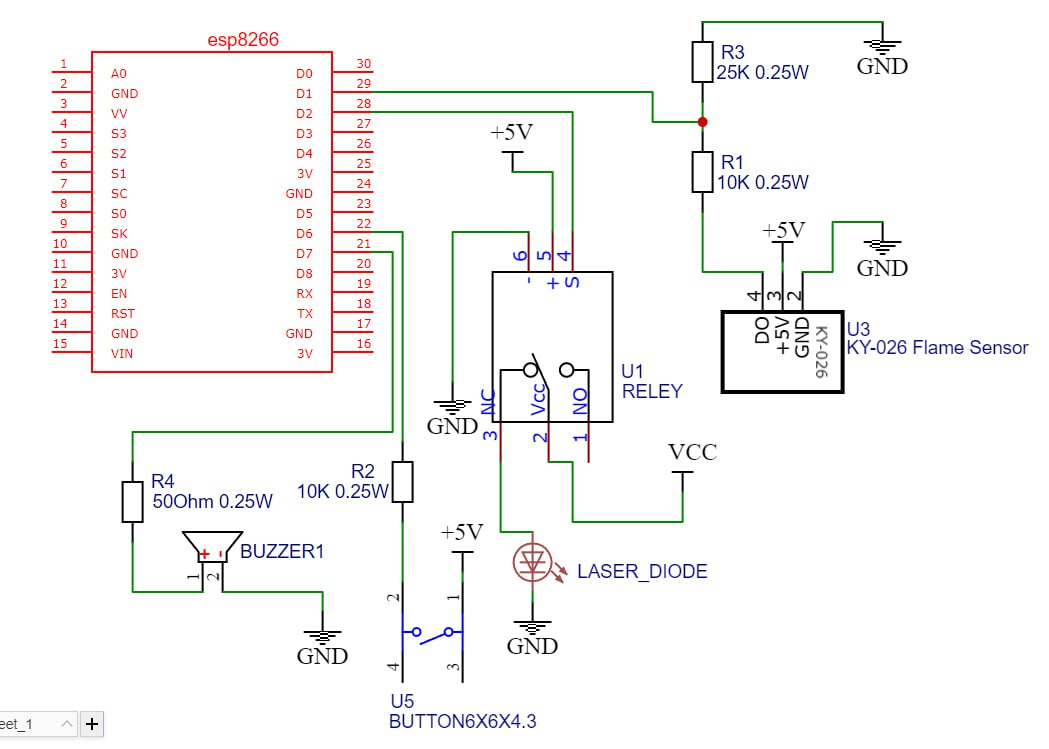


Рисунок 1.4 Схема устройства.

# 2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для разработки программного обеспечения данного курсового проекта будет использоваться язык C++, IDE Visual Studio Code вместе с расширением PlatformIO.

## 2.1 Язык программирования C++ история, версии, достоинства

**2.1.1** C++ был разработан в начале 1980-х годов Бьёрном Страуструпом в качестве расширения языка C для создания более мощного инструмента программирования, поддерживающего объекты и классы. Первая версия языка была представлена в 1985 году. C++ быстро стал популярным благодаря сочетанию возможностей структурного и объектно-ориентированного программирования.

**2.1.2** 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.

**2.1.3** Высокая производительность – C++ предоставляет низкоуровневый контроль над памятью и аппаратными ресурсами, что позволяет создавать быстрые и эффективные программы.

* Объектно-ориентированное программирование (ООП) – поддержка классов и объектов позволяет создавать масштабируемые и гибкие приложения.
* Широкие возможности стандартной библиотеки (STL) – набор контейнеров, алгоритмов и функций упрощает разработку.
* Совместимость с C – C++ является расширением C, что позволяет использовать код, написанный на C, в программах на C++.
* Гибкость и кроссплатформенность – программы на C++ могут быть скомпилированы и выполнены на различных операционных системах и платформах.

## 2.2 Visual Studio Code история, версии, достоинства

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

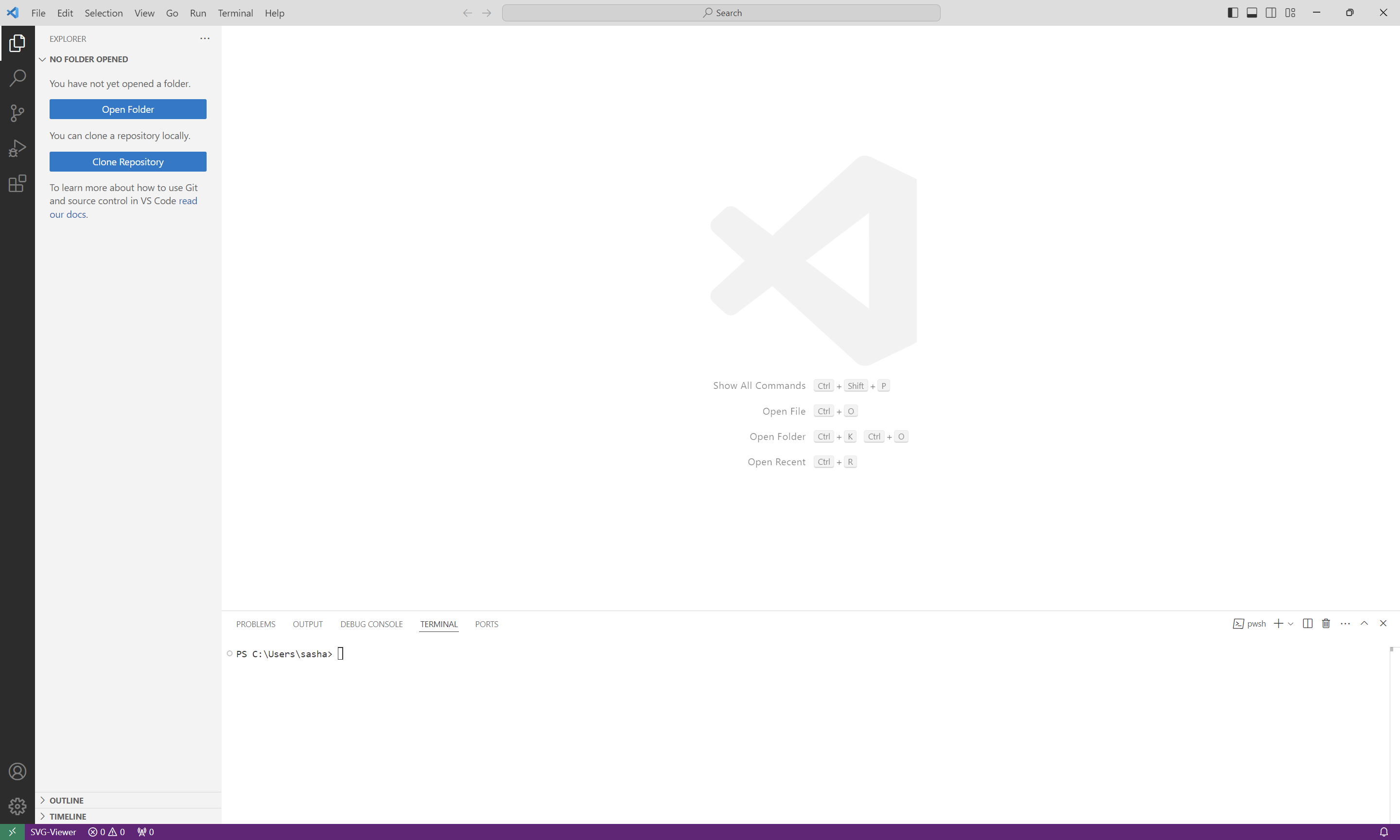


Рисунок 2.2 Интерфейс Visual Studio Code.

**2.2.1** 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.

**2.2.2** 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.

**2.2.3** 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.

## 2.3 PlatformIO история, версии, достоинства

**2.3.1** 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.

**2.3.2** 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.

**2.3.3** 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.

# ​СПИСОК ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

[1] Сайт производителя ESP32 [Электронный ресурс]. – Режим доступа: <https://www.espressif.com> – Дата доступа: 29.09.2024.

[2] Flammable Gas Sensor（Model：MQ-2）/ Zhengzhou Winsen Electronics Technology Co, 2020. – 7 с.

[3] Модуль интерфейсов BLIIoT M160T [Электронный ресурс]. – Режим доступа: <https://www.bliiot.com/mqtt-io-modules-p00258p1.html> – Дата доступа: 29.09.2024

[4] Описание протокола Modbus TCP [Электронный ресурс]. – Режим доступа: <https://ipc2u.ru/articles/prostye-resheniya/modbus-tcp/> – Дата доступа: 29.09.2024

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