

Разработка системы контроля качества воздуха на рабочем месте с паяльной станцией и уведомлением пользователя

Плюснина Софья Васильевна
Выручаев Дмитрий Алексеевич

Проблемы

1

Здоровье и
безопасность
студентов.

2

Недостаточная
осведомленность.

3

Недостаток
мониторинга и
контроля.

Описание проекта

Разработка системы контроля качества воздуха на рабочем месте с паяльной станцией и уведомлением пользователя помогает решить эти проблемы. Она предупреждает студентов о повышенном уровне дыма и загрязнения воздуха, а также предоставляет информацию о необходимости перерыва или отключения паяльной станции. Это позволяет сохранять здоровье и безопасность студентов в студенческих лабораториях и снижает риски, связанные с работой в загрязненной среде.

Аналоги проектов



Перечень комплектующих



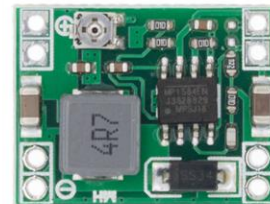
Паяльник



Блок питания

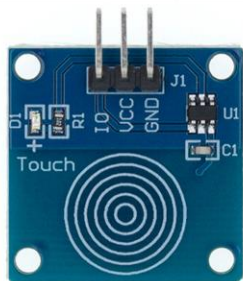


Модуль SONGLE
SRD-05VDC-SL-C



DC-DC
ПРЕОБРАЗОВАТЕЛЬ

Перечень комплектующих



СЕНСОРНЫЙ
КНОПОЧНЫЙ МОДУЛЬ



WEMOS D1 MINI



СВЕТОДИОД

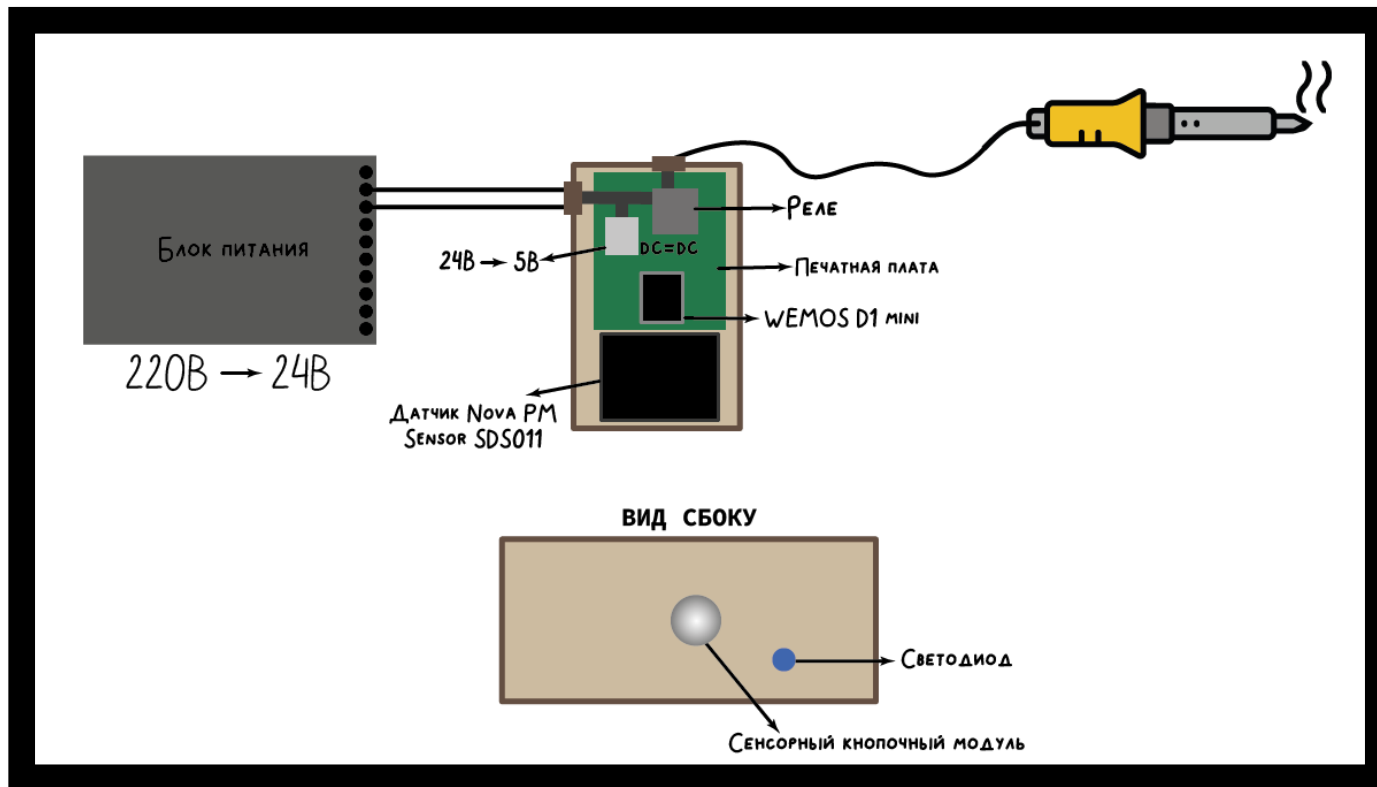


ДАТЧИК NOVA PM
SENSOR SDS011

Перечень комплектующих

- Блок питания → предоставляет электроэнергию всем компонентам системы
- DC-DC ПРЕОБРАЗОВАТЕЛЬ → преобразование напряжения из блока питания в нужное напряжение
- РЕЛЕ SONGLE SRD-05VDC-SL-C → для включения и отключения паяльной станции
- СЕНСОРНЫЙ КНОПОЧНЫЙ МОДУЛЬ → для управления реле
- WEMOS D1 MINI → управление системой контроля качества воздуха
- СВЕТОДИОД → оповещает о необходимости на время прекратить работу
- Датчик NOVA PM SENSOR SDS011 → для измерения концентрации частиц воздуха определенного размера (PM2.5 и PM10)

Макет паяльной станции



Сценарий использования устройства

01

Подключение
станции в
розетку

02

Вход в
приложение с
помощью
сканирования
QR-кода

03

Главная
страница с
информацией

04

Управление
паяльником

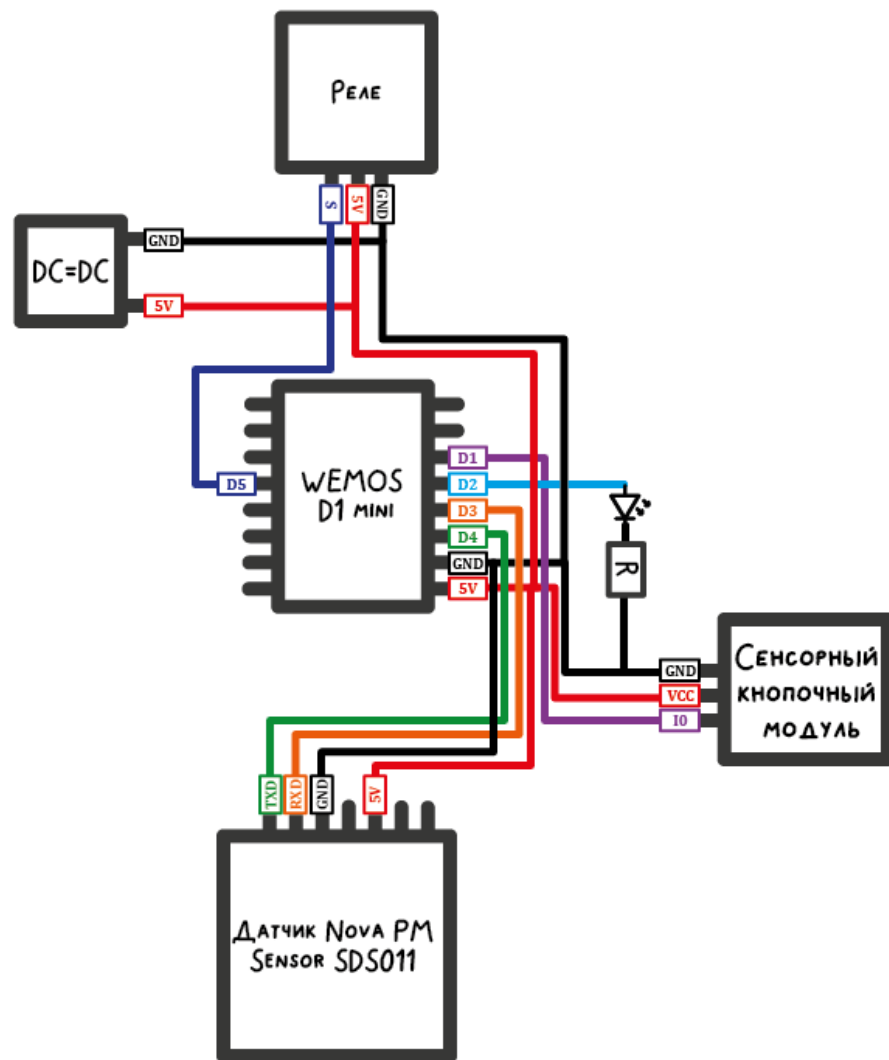
05

Отображение
показателей
качества
воздуха

06

Выход из
системы через
выключение
паяльника или
блока питания

Схема подключения



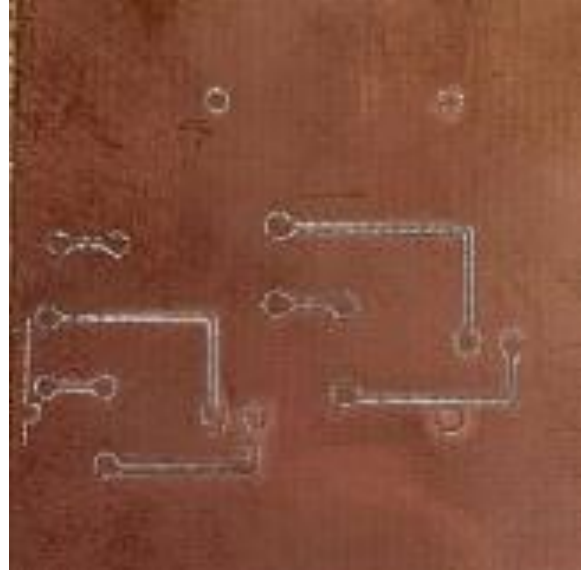
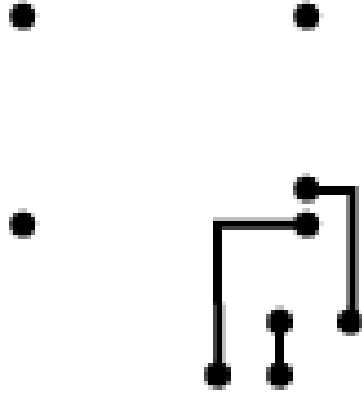
Печатная плата



Фрезерный станок
Стеклотекстолит

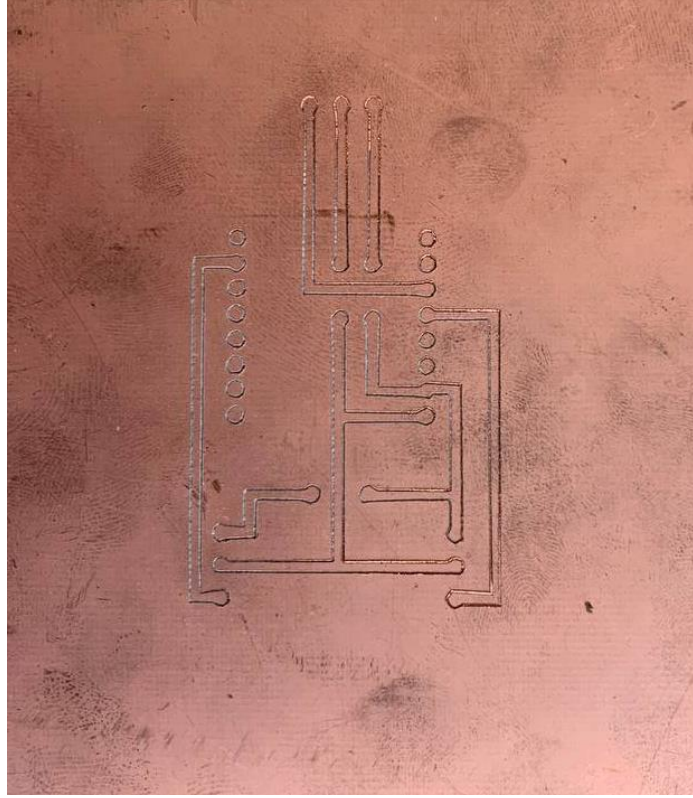
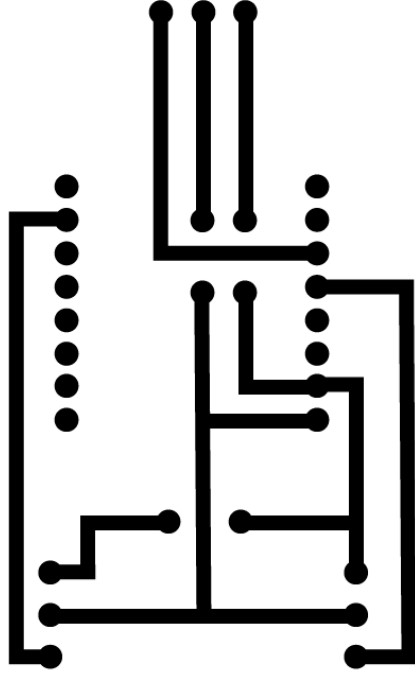
Печатная плата

Первая проба



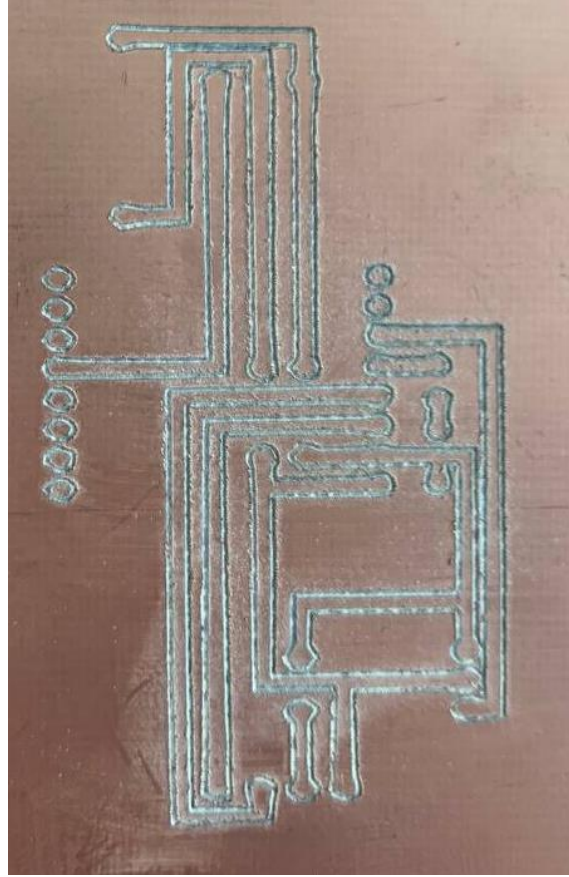
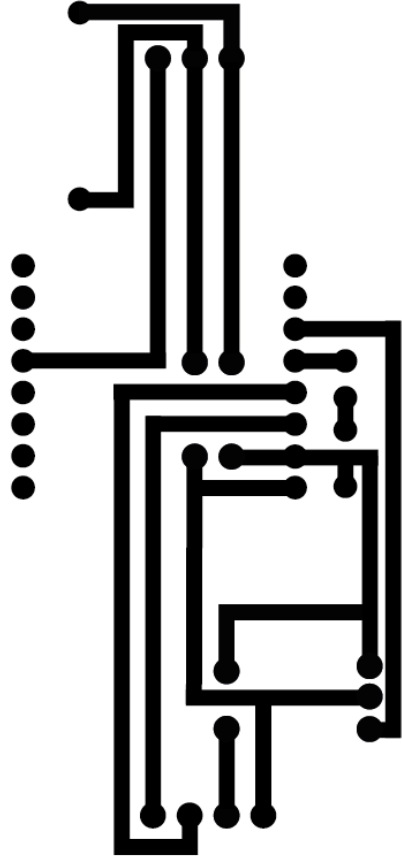
Печатная плата

Вторая проба



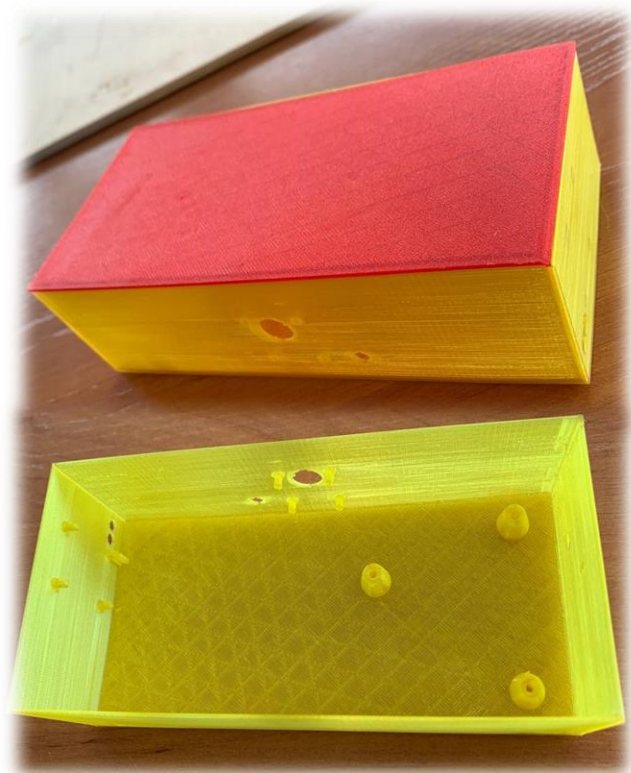
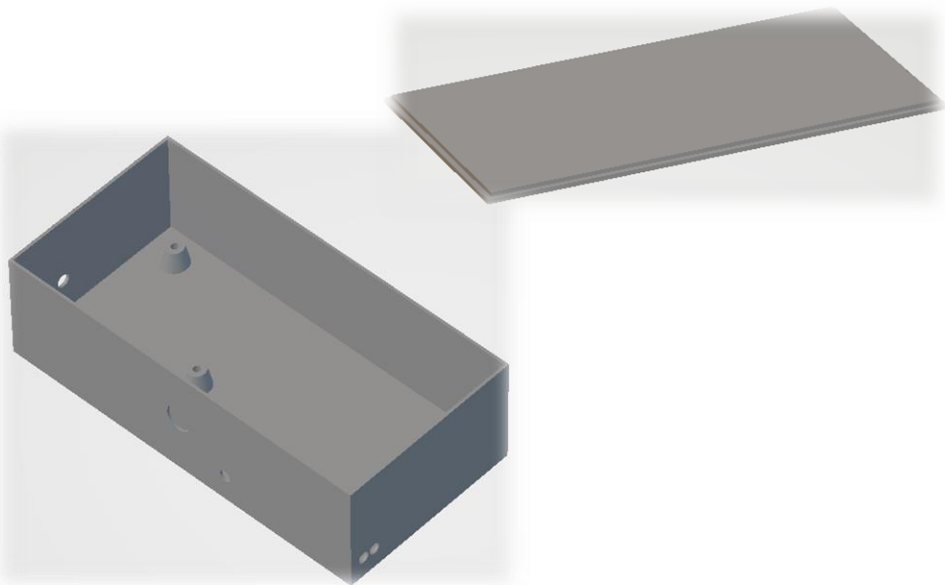
Печатная плата

Третья проба



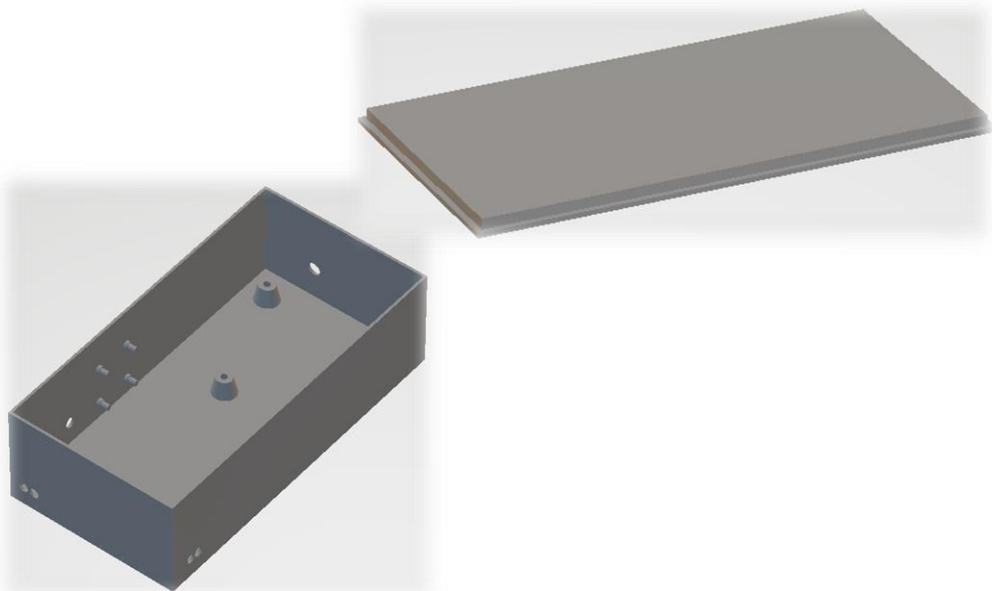
ЗД-моделирование и печать корпуса устройства

Первая проба



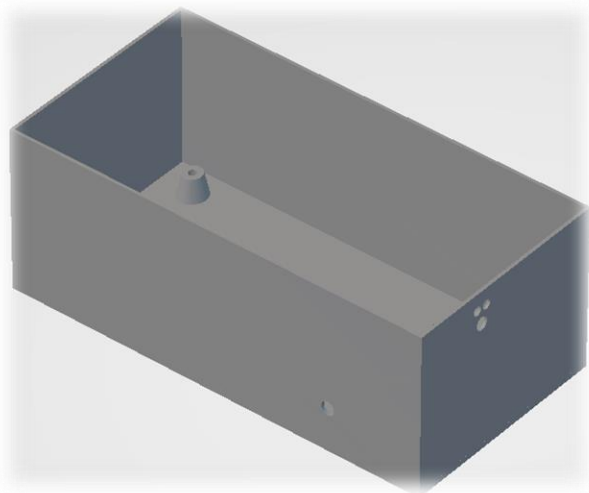
ЗД-моделирование и печать корпуса устройства

Вторая проба



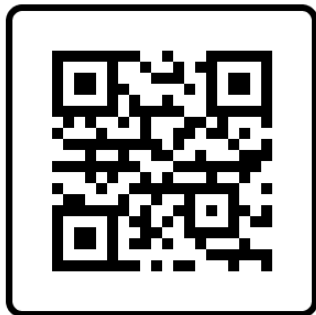
ЗД-моделирование и печать корпуса устройства

Третья проба



Варфрейм приложения

Сканировать



Главная

Уведомления



главная

паяльник

воздух

Паяльник

Вкл/
Выкл

главная

паяльник

воздух

Воздух

Текущие данные



Предыдущие данные

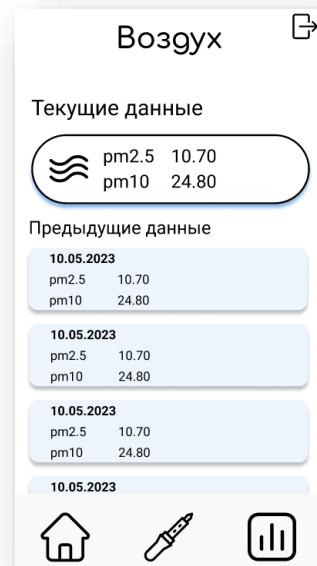
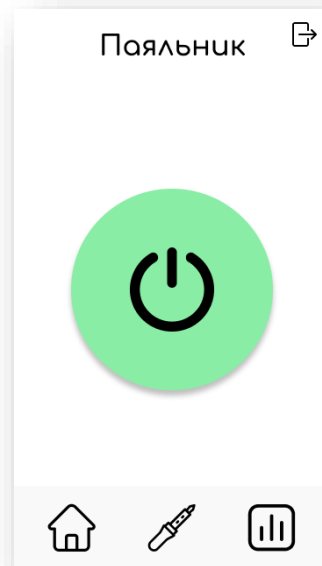
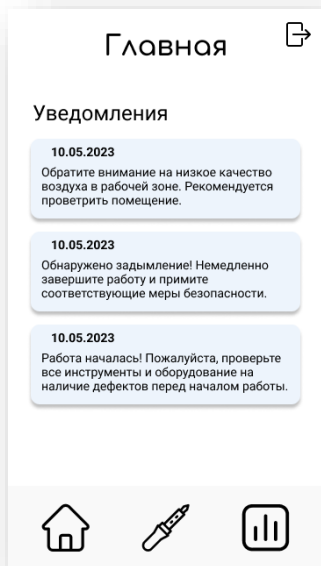


главная

паяльник

воздух

Дизайн приложения



Посмотреть приложение

Демонстрация дизайна приложения

<https://youtube.com/shorts/8n49v2c4n3w?feature=share>



Сценарий использования приложения

Запуск приложения

Пользователь запускает мобильное приложение на своем устройстве Android.

Вход в систему

Пользователь сканирует QR-код на рабочем месте с помощью приложения, после чего система генерирует и присваивает уникальный токен пользователю.

Главная страница

После входа в систему пользователь видит на главной странице уведомления о состоянии воздуха и другие важные сообщения.

Управлением паяльником

В этой секции пользователь может включать и выключать паяльное устройство удаленно с помощью соответствующей кнопки.

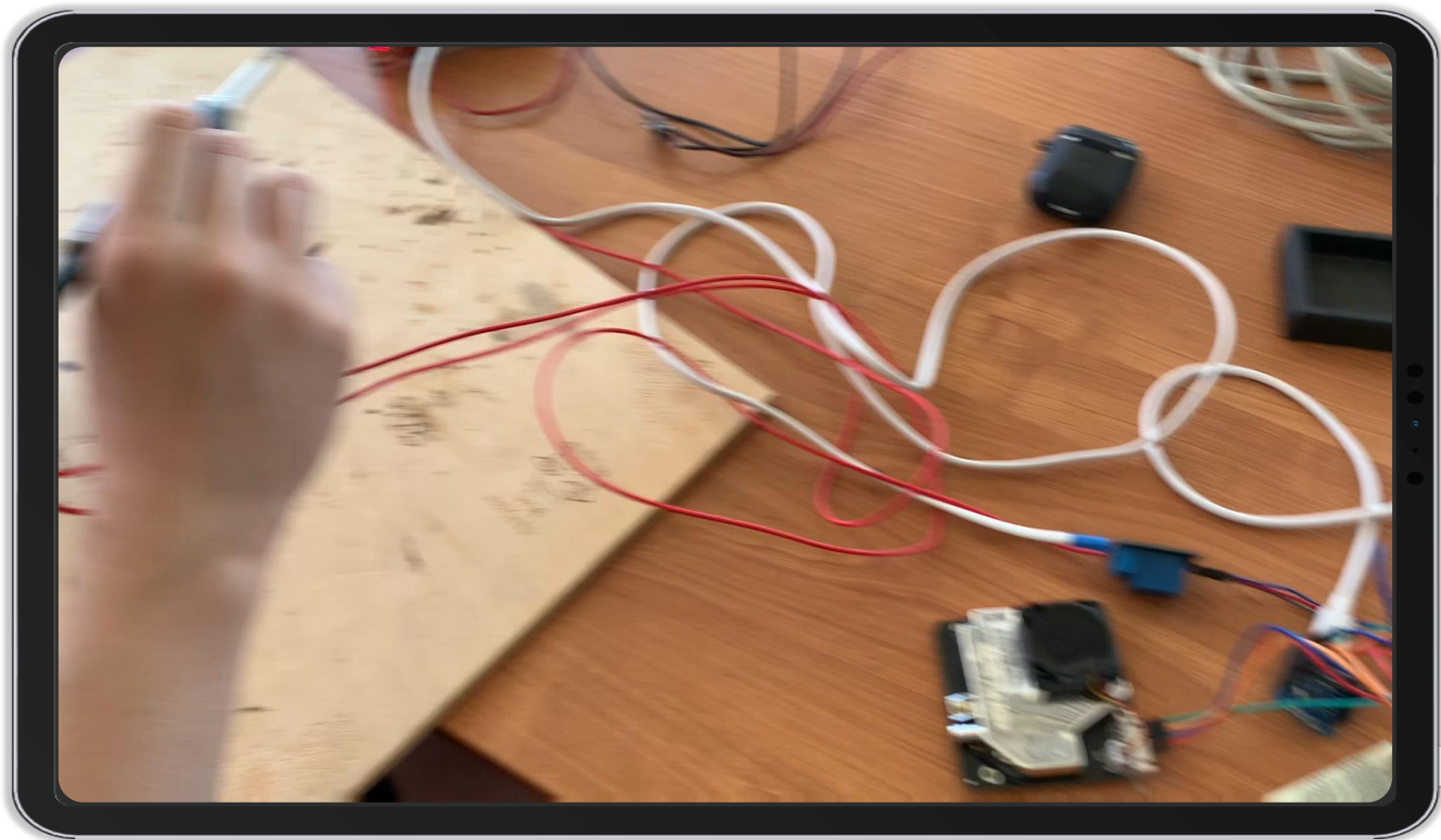
Показатели воздуха

Эта секция отображает текущие показатели качества воздуха на рабочем месте, полученные от датчика.

Выход в системы

Пользователь может выйти из системы, нажав на значок выхода. После выхода пользователь теряет доступ к уведомлениям и удаленному управлению паяльником.

Демонстрация работы



<https://youtu.be/JIQZ7QGF9jc>

Код

```
#include <SoftwareSerial.h>

SoftwareSerial sdsSerial(D4, D3); // Подключение датчика SDS011 к пинам D4 (RX) и D3 (TX) на плате Wemos D1 Lite mini

const int ledPin = D2;          // Пин подключения светодиода
const int buttonPin = D1;       // Пин подключения кнопки TTP223
const int relayPin = D5;        // Пин подключения реле

bool buttonState = false;
bool previousButtonState = false;
bool relayState = false;
int sensorValue = 0;

void setup() {
    pinMode(ledPin, OUTPUT);      // Установка пина светодиода в режим вывода
    pinMode(buttonPin, INPUT_PULLUP); // Установка пина кнопки в режим входа с подтяжкой вверх
    pinMode(relayPin, OUTPUT);    // Установка пина реле в режим вывода

    digitalWrite(relayPin, LOW); // Изначально реле выключено

    Serial.begin(9600);           // Инициализация последовательного порта
    sdsSerial.begin(9600);        // Инициализация SoftwareSerial для работы с датчиком SDS011
}
```


Код

```
void loop() {
  nova_sensor(); // Считывание данных с датчика

  // Управление светодиодом
  if (sensorValue > 35000) {
    digitalWrite(ledPin, HIGH); // Включение светодиода, если значение с датчика выше 35000
  } else {
    digitalWrite(ledPin, LOW); // Выключение светодиода в остальных случаях
  }

  buttonState = digitalRead(buttonPin); // Считывание состояния кнопки

  if (buttonState != previousButtonState) { // Если состояние кнопки изменилось
    if (buttonState == LOW) { // Если кнопка нажата (логический уровень LOW)
      Serial.println("Button pressed!");
      relayState = !relayState; // Инвертирование состояния реле

      // Управление состоянием реле
      if (relayState) {
        digitalWrite(relayPin, HIGH); // Включение реле
        Serial.println("Relay turned ON!");
      } else {
        digitalWrite(relayPin, LOW); // Выключение реле
        Serial.println("Relay turned OFF!");
      }
    }
  }

  previousButtonState = buttonState; // Обновление предыдущего состояния кнопки
  delay(1000); // Задержка для стабилизации считывания состояния кнопки
}
```

Код

```
void nova_sensor() {  
  if (sdsSerial.available()) {  
    if (sdsSerial.find(0xAA)) {    // Поиск начала пакета данных  
      byte data[10];  
      if (sdsSerial.readBytes(data, 10) == 10) {  
        // Извлечение данных из пакета  
        int pm25 = (data[3] * 256 + data[2]);    // Значение PM2.5  
        int pm10 = (data[5] * 256 + data[4]);    // Значение PM10  
  
        // Вывод данных в монитор порта  
        Serial.print("PM2.5: ");  
        Serial.print(pm25);  
        Serial.print(" ug/m3, PM10: ");  
        Serial.print(pm10);  
        Serial.println(" ug/m3");  
  
        // Проверка значения PM2.5  
        if (pm25 > 35000) {  
          sensorValue = pm25;  
        } else {  
          sensorValue = 0;  
        }  
      }  
    }  
  }  
  
  delay(100); // Пауза 1 секунда между измерениями  
}
```

Спасибо за внимание!