

Универзитет “Св. Кирил и Методиј” – Скопје

Проект по предмет:

Вградливи микропроцесорски системи

AeroSense – Control your home with a click of a button

Предметен професор/ментор:

Невена Ацковска

Изработено од:

Слободанка Пиштова

211171

Table of Contents

| | |
|---------------------------------------|----|
| <i>Вовед</i> | 3 |
| Фази на креирање на апликацијата..... | 4 |
| Фаза 1: Планирање и анализа | 4 |
| Фаза 2: Дизајнирање..... | 6 |
| Фаза 3: Имплементација | 9 |
| <i>Фаза 4: Продукција</i> | 13 |
| Заклучок..... | 14 |

Анстракт

Во ера каде технологијата беспрекорно се интегрира во секој аспект од нашите животи, нашите домови еволуираа во паметни екосистеми дизајнирани да ја подобрат удобноста и благосостојбата. Од паметни термостати до автоматизирани системи за осветлување, модерното домаќинство е опремено со алатки кои го прават секојдневниот живот поефикасен и попријатен.

Меѓутоа, како што стануваме се поприлагодени на придобивките од овие технологии, важноста од одржување на оптимална внатрешна средина е клучна за здравјето, продуктивноста и севкупниот квалитет на живот.

Со креирање на алатката *AeroSense*, создадов решение за следење и управување со квалитетот на воздухот во затворените простории, температурата и влажноста, се во една апликација.

Вовед

Секојдневно се соочуваме со сè поекстремни температури и зголемено ниво на загаденост на воздухот. Потребата да го заштитиме нашето здравје и да ја одржуваме контролата врз нашите внатрешни средини никогаш не била покритична.

AeroSense стапува како витално решение за овие современи предизвици. Тој нуди напредно следење и управување со квалитетот на воздухот во затворените простории, температурата и влажноста, обезбедувајќи увид во реално време и автоматска контрола за да ја одржувате вашата околина безбедна и удобна.

AeroSense ви дава можност да преземете проактивни мерки против ефектите од екстремните временски услови и загадениот воздух, осигурувајќи дека вашите внатрешни простори остануваат здрави и добро регулирани, без разлика на надворешните услови.

Во време кога нашите домови стануваат попаметни и поинтегрирани, *AeroSense* се издвојува како клучна алатка за одржување на здрава животна средина. Неговата способност за следење, прилагодување и оптимизирање на условите во затворен простор не само што ја оптимизира удобноста, туку и игра клучна улога во заштитата на нашето здравје и благосостојба.

Фази на креирање на апликацијата

Фаза 1: Планирање и анализа

1) *Потреба (Need)*

Во денешниот свет, каде климатските промени и урбанизацијата доведуваат до поекстремни временски услови и повисоки нивоа на загадување на воздухот, одржувањето на здрава внатрешна средина станува сè поголем предизвик.

Лошиот квалитет на воздухот во затворените простории и екстремните температури може да имаат сериозни ефекти врз здравјето, вклучувајќи респираторни проблеми, алергии и општа непријатност.

Традиционалните системи за греење, вентилација и климатизација (HVAC) често немаат интелигенција за да се прилагодат на променливите услови во реално време, што резултира со неефикасно користење на енергија и неоптимални услови за живеење.

AeroSense го решава овој проблем со обезбедување на паметно, автоматизирано решение кое континуирано нуди контрола на нашата околина и оптимизира квалитетот на воздухот во затворените простории, температурата и влажноста, обезбедувајќи поздрава и поудобна животна средина.

2) *Целна група (Target Audience)*

Овде може да се напоменат:

- 1) **Сопственици на домови:** Поединци и семејства кои сакаат да обезбедат здрава и удобна животна средина, особено во региони со екстремни температури или високо ниво на загадување.
- 2) **Жителите на станови:** Жителите на урбаните области каде што квалитетот на воздухот е често загрозен и кои може да имаат ограничена контрола врз системите за климатизација и климатизација на нивната зграда.
- 3) **Здравствени свесни поединци:** Луѓе со респираторни состојби, алергии или други здравствени проблеми што ги прават особено чувствителни на квалитетот на воздухот и температурните флуктуации.
- 4) **Ентузијастички за паметни домови:** Корисници кои се запознаени со технологијата кои бараат да интегрираат напредни еколошки контроли во нивните екосистеми за паметни домови.

- 5) **Мали бизниси:** Мали канцеларии, малопродажни простори и други комерцијални средини кои треба да одржуваат оптимални услови за благосостојба на вработените и удобност на клиентите.
- 6) **Сопственици на оранжерии, градини и слично:** Поединци или бизниси вклучени во земјоделство или градинарство во затворени простории кои бараат прецизна контрола врз условите на животната средина за здравјето на растенијата.

3) Побарувачка (Requirements)

Системот AeroSense треба да исполни неколку клучни барања за да биде ефективен и лесен за користење:

- 1) **Интеграција на сензори:** Системот треба да се интегрира со повеќе сензори, вклучувајќи ги и оние за температура, влажност, квалитет на воздухот (на пример, VOCs, нивоа на CO₂) и евентуално нивоа на светлина и бучава.
- 2) **Автоматизација и контрола:** Мора да биде способен да контролира HVAC системи, вентилатори, навлажнувачи и други уреди за контрола на животната средина врз основа на податоците од сензорот.
- 3) **Кориснички предупредувања:** AeroSense треба да обезбеди предупредувања и известувања за критични промени во животната средина, како што се скокови на температурата или слаб квалитет на воздухот, за да поттикне итна акција.
- 4) **Приспособливост:** AeroSense треба да биде скалабилен, дозволувајќи им на корисниците да додаваат повеќе сензори и контролни единици по потреба за покривање на поголеми или посложени средини.
- 5) **Евиденција и анализа на податоци (опционално):** Системот треба да ги евидентира податоците за животната средина со текот на времето, овозможувајќи им на корисниците да ги анализираат трендовите и да донесуваат информирани одлуки.
- 6) **Далечинско следење (опционално):** AeroSense треба да им овозможи на корисниците да ја следат и контролираат својата внатрешна околина од далечина преку мобилна апликација или веб-интерфејс.

4) Конкуренција (Competitors)

Неколку конкуренти нудат паметни производи за нашиот дом и просторот за следење на животната средина. Тие вклучуваат:

- 1) **Nest Thermostat (од Google):** Широко користен паметен термостат што ги учи корисничките параметри и ја оптимизира контролата на температурата. Сепак, тој првенствено се фокусира на температурата и не нуди сеопфатен мониторинг на квалитетот на воздухот.
- 2) **Ecobee SmartThermostat:** Друг популарен паметен термостат кој вклучува сензори за температура и се интегрира со системи за паметни домови како Alexa. Нуди одредено следење на животната средина, но нема опсежни контроли за квалитетот на воздухот.
- 3) **Awair Element:** Наменски монитор за квалитетот на воздухот што ја следи температурата, влажноста, нивото на CO₂, хемикалииите (VOCs) и ситната прашина (PM_{2,5}). Иако се истакнува во следењето на квалитетот на воздухот, не ги контролира директно системите за климатизација и климатизација.
- 4) **Dyson Pure Cool:** Комбинација од прочистувач на воздух и вентилатор што го следи квалитетот на воздухот и го прилагодува протокот на воздух за да го подобри воздухот во затворените простории. Сепак, тој е ограничен на прочистување на воздухот и не се интегрира со постоечките системи за климатизација и климатизација или нуди сеопфатни еколошки контроли.
- 5) **Netatmo Healthy Home Coach:** паметен уред кој го следи квалитетот на воздухот во затворените простории, влажноста, температурата и нивото на бучава. Обезбедува предупредувања и препораки, но не ги автоматизира системите за контрола на животната средина.

AeroSense се разликува со комбинирање на сеопфатно следење на животната средина со паметна автоматизација и контрола, обезбедувајќи поинтегриран и поедноставен сеопфатен пристап за одржување здрави внатрешни средини.

Фаза 2: Дизајнирање

1) Архитектура

За да се креира AeroSense, архитектурата на системот мора внимателно да се планира за да се обезбеди непрекорна интеграција на хардвер, софтвер и комуникациски компоненти. Архитектурата може да се подели на неколку клучни слоеви и компоненти, од кои секоја игра клучна улога во целокупната функционалност на системот.

1) Слој на сензори

Слојот на сензори е одговорен за собирање податоци за животната средина како температура, влажност и квалитет на воздухот. Овој слој вклучува различни сензори и нивните интерфејси со Arduino.

Клучни компоненти:

Сензори DHT22/DHT11: Овие сензори ја мерат температурата и влажноста. Тие се поврзани со Arduino преку дигитални пинови, обезбедувајќи точни читања на околината.

Сензор за квалитет на воздух MQ-135: Овој сензор детектира штетни гасови како амонијак, сулфур, бензен пареа, чад и CO₂. Комуницира со Arduino користејќи аналогни пинови за испраќање на напонски сигнал што одговара на концентрацијата на гасовите во воздухот.

Сензор за светлина (LDR) (опционално): LDR (Light Dependent Resistor) се користи за мерење на нивото на амбиентална светлина. Овој сензор е исто така поврзан со аналогните

Сензор за влага во почвата (опционално): За AeroSense да се прошири за да вклучи следење на здравјето на растенијата, може да се додаде сензор за влага во почвата за мерење на содржината на вода во почвата.

Анемометар и ветровитарка (опционално): Тие се користат за мерење на брзината и насоката на ветерот доколку системот е распореден на отворено.

2) Контролен слој

Контролниот слој управува со интеракцијата помеѓу сензорите и актуаторите кои ја прилагодуваат околината врз основа на добиените податоци.

Клучни компоненти:

Модули за пренесување информации (relay modules) : Тие се користат за контрола на уреди со голема моќност како вентилатори, грејачи, навлажнувачи и прочистувачи на воздух. Тие делуваат како прекинувачи кои се активираат од Arduino врз основа на читањата на сензорот.

Контрола на PWM (Модулација на ширина на пулсот): PWM се користи за контрола на уреди како вентилатори или LED светла, каде брзината или осветленоста треба постепено да се прилагодуваат. Arduino може да ја модулира енергијата што се снабдува на овие уреди користејќи PWM сигнали.

3) Процесиращки слој

Овој слој се справува со логичките процеси и процесите на донесување одлуки кои одредуваат како системот реагира на податоците од сензорот, и овде ја користиме нашиот клучна Arduino компонента која ќе ни помогне да го направиме овој проект во реалност.

Клучни компоненти:

Микроконтролер Arduino (на пример, Arduino Uno, Nano или Mega): Централниот мозок на системот, одговорен за читање на податоците од сензорот, нивна обработка и активирање дејства врз основа на претходно дефинирани алгоритми.

Вграден софтвер: Софтверот што работи на Arduino, вообичаено напишан во C/C++ со помош на Arduino IDE, ја имплементира логиката за читање податоци од сензорот, донесување одлуки и контролирање на актуаторите.

- Покрај главните целини, може исто така да се напоменат и комуникациски слој, UI слој како и слој за складирање податоци, но овде ќе се фокусираме на работа со нашата Arduino компонента и пишување на софтвер т.е код што ќе биде имплементиран во истата.

2) Технички ресурси

Хардвер ресурси:

Arduino Board UNO: Основниот микроконтролер што ќе работи со системот AeroSense.

Сензор DHT22/DHT11: За мерење на температура и влажност.

Сензор за квалитет на воздух MQ-135: За откривање на штетни гасови и мерење на квалитетот на воздухот.

LDR (Light Dependent Resistor) (опционално): За мерење на нивоата на амбиентална светлина.

Relay модули: За контрола на уреди со голема моќност како вентилатори, грејачи, навлажнувачи и прочистувачи на воздух.

Уреди компатибилни со PWM: како што се вентилатори или LED светла за контрола на променлива брзина/осветленост.

5V/12V Напојување: За напојување на Arduino, сензори и актуатори.

Резервна копија на батеријата (опционално): За да продолжи системот да работи при прекини на струја.

LCD/OLED дисплеј (опционално): За локално прикажување во реално време на податоците од сензорот и статусот на системот.

Breadboard и Jumper жици: За прототипирање и поврзување на компоненти за време на развојот.

Модул за SD-картичка (опционално): За локално складирање на податоци на SD-картичка.

Софтверски ресурси:

Arduino IDE: Примарна развојна околина за пишување, компајлирање и поставување код на Arduino.

Ова ќе биде главната алатка што ќе ја искористиме за креирање на апликацијата.

Платформа за развој на мобилни апликации (опционално): Android Studio (за Android), Xcode (за iOS) или алатки за меѓуплатформи како Flutter или React Native за развој на мобилната апликација AeroSense.

Алатки за развој на веб апликација (опционално): HTML/CSS, Python, AWS

Фаза 3: Имплементација

За понатамошна изработка на нашиот проект, користиме Arduino Client и креираме нов Sketch каде што го поставуваме нашиот код. Бидејќи ни се потребни повеќе библиотеки за имплементација на нашите сензори (како на пример за DHT11 сензорот) одиме во Tools-> Manage Libraries и ја инсталираме DHT Sensor Library. Истото го извршуваме и за MQ135 сензорот што ни помага да го тестираме квалитетот на воздухот. Нашиот код со инсталираните библиотеки би изгледал вака

```

1  #include <DHT.h>
2  #include <MQ135.h>
3
4  // Pin Definitions
5  #define DHTPIN 2           // Digital pin connected to the DHT sensor
6  #define DHTTYPE DHT22     // DHT 22 (AM2302)
7  #define MQ135PIN A0       // Analog pin connected to the MQ135 sensor
8  #define RELAY_FAN 5       // Digital pin connected to the fan relay
9  #define RELAY_HEATER 4    // Digital pin connected to the heater relay
10 #define RELAY_VENT 6      // Digital pin connected to the ventilation system relay
11 #define BUZZER 7          // Digital pin connected to a buzzer
12
13 // Sensor Objects
14 DHT dht(DHTPIN, DHTTYPE);
15 MQ135 mq135(MQ135PIN);
16
17 // Variables to store sensor readings
18 float temperature;
19 float humidity;
20 int airQuality;
21
22 // Thresholds
23 const int airQualityThreshold1 = 150; // Threshold for moderate air quality (PPM)
24 const int airQualityThreshold2 = 280; // Threshold for poor air quality (PPM)
25
26 void setup() {
27   Serial.begin(9600);           // Start serial communication at 9600 baud
28   dht.begin();                 // Initialize the DHT22 sensor
29   pinMode(RELAY_FAN, OUTPUT);  // Set fan relay pin as output
30   pinMode(RELAY_HEATER, OUTPUT); // Set heater relay pin as output
31   pinMode(RELAY_VENT, OUTPUT);  // Set ventilation relay pin as output
32   pinMode(BUZZER, OUTPUT);     // Set buzzer pin as output
33   digitalWrite(RELAY_FAN, LOW); // Start with fan off
34   digitalWrite(RELAY_HEATER, LOW); // Start with heater off
35   digitalWrite(RELAY_VENT, LOW); // Start with ventilation system off
36   digitalWrite(BUZZER, LOW);   // Start with buzzer off
37 }
38
39 void loop() {
40   // Read Sensor Data
41   temperature = dht.readTemperature();
42   humidity = dht.readHumidity();
43   airQuality = mq135.getPPM(); // Get the PPM value from MQ135
44
45   // Print the sensor values to the Serial Monitor
46   Serial.print("Temperature: ");
47   Serial.print(temperature);
48   Serial.print(" °C, Humidity: ");
49   Serial.print(humidity);
50   Serial.print(" %, Air Quality: ");
51   Serial.print(airQuality);
52   Serial.println(" PPM");
53
54   // Control fan and heater based on temperature
55   if (temperature > 25.0) {
56     digitalWrite(RELAY_FAN, HIGH); // Turn on the fan
57     digitalWrite(RELAY_HEATER, LOW); // Turn off the heater
58   } else if (temperature < 18.0) {
59     digitalWrite(RELAY_HEATER, HIGH); // Turn on the heater
60     digitalWrite(RELAY_FAN, LOW); // Turn off the fan
61   } else {
62     digitalWrite(RELAY_FAN, LOW); // Turn off the fan
63     digitalWrite(RELAY_HEATER, LOW); // Turn off the heater
64   }
65
66   // Control ventilation and alarm based on air quality
67   if (airQuality > airQualityThreshold2) {
68     digitalWrite(RELAY_VENT, HIGH); // Turn on the ventilation system
69     digitalWrite(BUZZER, HIGH); // Turn on the buzzer to signal very poor air quality
70     // Additional actions can be added here
71   } else if (airQuality > airQualityThreshold1) {
72     digitalWrite(RELAY_VENT, HIGH); // Turn on the ventilation system
73     digitalWrite(BUZZER, LOW); // Keep the buzzer off for moderate air quality
74   } else {
75     digitalWrite(RELAY_VENT, LOW); // Turn off the ventilation system
76     digitalWrite(BUZZER, LOW); // Turn off the buzzer
77   }
78
79   // Wait for a minute before next reading
80   delay(60000);
81 }

```

Дел по дел може да го разгледаме кодот.

Void setup() - Кодот го иницијализира Arduino да работи со сензорот DHT22 (за температура и влажност) и сензорот MQ-135 (за квалитет на воздухот). Исто така, поставува два дигитални пинови за контролирање на вентилаторот и грејачот со помош на модули

Void loop() - Кодот постојано ги чита податоците за температурата и влажноста од сензорот DHT22 и податоците за квалитетот на воздухот од сензорот MQ-135. Читањата на сензорот се печатат на серискиот монитор, за да можете да се видат во реално време (RTC).

Контролна логика (if циклус бр. 1) - Врз основа на температурните отчитувања:

Ако температурата е над 25°C, вентилаторот се вклучува за да се излади околината.

Ако температурата е под 18°C, грејачот се вклучува за да се загрее околината.

Ако температурата е помеѓу 18°C и 25°C, и вентилаторот и грејачот остануваат исклучени.

Контролна логика (if циклус бр. 2) - Врз основа на квалитетот на воздухот:

Ако квалитетот на воздухот ја преминува вредноста за лош квалитет, се уклучува вентилациониот систем и се уклучува алармот.

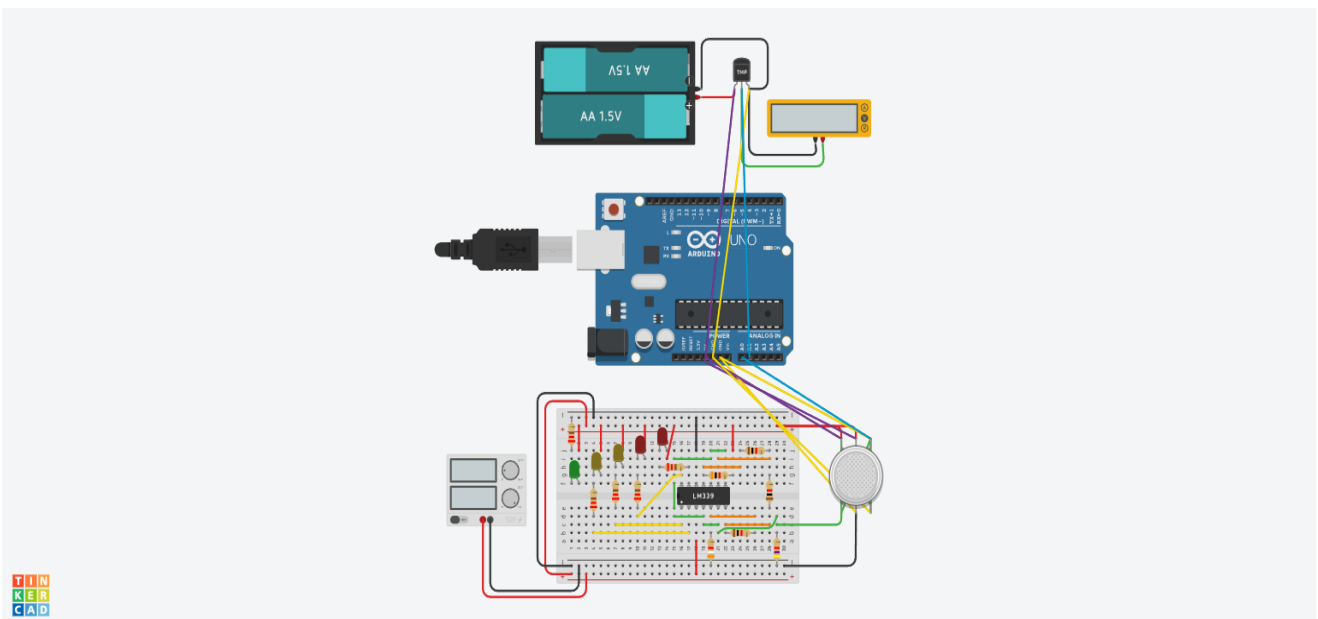
Ако квалитетот на воздухот ја преминува вредноста за нормален квалитет на воздухот но помала или еднаква на оптималната вредност, се уклучува вентилација но не се уклучува алармот.

Ако квалитетот на воздухот ја достигнал оптималната вредност, не се уклучува ниту вентилациониот систем ниту алармот.

Чекање (delay делот) :

По секој сет на читања и контролни дејства, кодот чека 60 секунди (60000 милисекунди) пред да го повтори процесот.

Дијаграмот би изгледал вака



Искористени се prebuilt аналогни компоненти за температура и детекција на квалитетот на воздухот. Што всушност имаме направено овде?

Температура:

Искористивме почетен (starter) што е поврзан со две AA батерии со по 1.5V поединечно. Имаме термометар (TMP компонента) со 3 метални делови од коишто поврзавме повеќе жици. Исто така имаме поврзано и мултиметар (жолтата компонента) за мерење на самата волтажа.

Како што напоменавме, TMP компонентата има 3 метални делови: Power (VCC) , Vout (OUT/SIGNAL) и Ground (GND). Како ги поврзавме овие делови?

Жиците се одбележани со редослед:

Сина: OUTPUT/SIGNAL -> A1

Виолетова: VCC/ -> 5V

Жолта: GND -> GND, дообјаснето

- VCC делот кај термометарот го поврзавме со 5V на нашето Arduino
- OUT делот кај термометарот го поврзавме со A1 на нашето Arduino (pin за излез)
- GND делот кај термометарот го поврзавме соодветно со GND делот на нашето Arduino

Квалитет на воздух:

Искористивме почетен аналоген гас сензор (кружен сив сензор со шест метални делови) што е поврзан со мал breadboard (самата плоча), неколку ЛЕД сијалички (зелени и црвени на сликата), резистори (мали жолти цилиндри по самата плоча) како и напојувач (power supply, лево со две жици). Главно ќе се фокусираме на нашиот гас сензор.

Веќе спомнавме дека има 6 метални делови од коишто може да се поврзат жици. Тие се A1, A2, B1, B2, H1 и H2. Секоја од нив има своја улога. Како ги поврзавме овие делови?

Жиците се одбележани со редослед:

Сина: A2 -> A0

Виолетова: A1/H1 -> 5V

Зелена: меѓусебно поврзување A1->B1, A2->B2

Жолта: A2 -> GND, дообјаснето:

- A1 – A1 е меѓусебно поврзана со B1 и таа е поврзана со 5V на нашето Arduino
- H1 – Исто како и A1, H1 е поврзано со 5V на нашето Arduino
- A2 – има 3 жици, поврзана е со B2, со GND на нашето Arduino и исто така со A0

- B2 – поврзана со A2 и исто така со GND на нашето Arduino
- H2 – поврзана стриктно со GND на нашето Arduino
- B1 – поврзана со A2 и A1

Со ова го изградивме нашиот Aerosense систем кој понатаму може да се подобрува на различни начини (User Interface, Database and Cloud Saving итн.)

Фаза 4: Продукција

Со неколку наведени промени, ние може да го ставиме самиот проект во продукција. Симулација на самиот проект беше изработена на Tinkercad каде што некои делови од кодот беа заменети поради фактот што Tinkercad не ги имаше потребните библиотеки за нашиот оригинален код.

```

1 // Pin Definitions
2 #define RELAY_FAN 5 // Digital pin connected to the fan relay
3 #define RELAY_HEATER 4 // Digital pin connected to the heater relay
4 #define RELAY_VENT 6 // Digital pin connected to the ventilation system relay
5 #define BUZZER 7 // Digital pin connected to a buzzer
6
7 // Variables to store sensor readings
8 float temperature;
9 float humidity;
10 int airQuality;
11
12 // Thresholds
13 const int airQualityThreshold1 = 150; // Threshold for moderate air quality (PPM)
14 const int airQualityThreshold2 = 200; // Threshold for poor air quality (PPM)
15
16 void setup() {
17   Serial.begin(9600); // Start serial communication at 9600 baud
18   pinMode(RELAY_FAN, OUTPUT); // Set fan relay pin as output
19   pinMode(RELAY_HEATER, OUTPUT); // Set heater relay pin as output
20   pinMode(RELAY_VENT, OUTPUT); // Set ventilation relay pin as output
21   pinMode(BUZZER, OUTPUT); // Set buzzer pin as output
22   digitalWrite(RELAY_FAN, LOW); // Start with fan off
23   digitalWrite(RELAY_HEATER, LOW); // Start with heater off
24   digitalWrite(RELAY_VENT, LOW); // Start with ventilation system off
25   digitalWrite(BUZZER, LOW); // Start with buzzer off
26 }
27
28 void loop() {
29   // Simulate sensor readings
30   temperature = random(15, 30); // Random temperature between 15 and 30°C
31   humidity = random(30, 70); // Random humidity between 30% and 70%
32   airQuality = random(100, 250); // Random air quality between 100 and 250 PPM
33
34   // Print the sensor values to the Serial Monitor
35   Serial.print("Temperature: ");
36   Serial.print(temperature);
37   Serial.print(" °C, Humidity: ");
38   Serial.print(humidity);
39   Serial.print(" %, Air Quality: ");
40   Serial.print(airQuality);
41   Serial.println(" PPM");
42
43   // Control fan and heater based on temperature
44   if (temperature > 25.0) {
45     digitalWrite(RELAY_FAN, HIGH); // Turn on the fan
46     digitalWrite(RELAY_HEATER, LOW); // Turn off the heater
47   } else if (temperature < 18.0) {
48     digitalWrite(RELAY_HEATER, HIGH); // Turn on the heater
49     digitalWrite(RELAY_FAN, LOW); // Turn off the fan
50   } else {
51     digitalWrite(RELAY_FAN, LOW); // Turn off the fan
52     digitalWrite(RELAY_HEATER, LOW); // Turn off the heater
53   }
54
55   // Control ventilation and alarm based on air quality
56   if (airQuality > airQualityThreshold2) {
57     digitalWrite(RELAY_VENT, HIGH); // Turn on the ventilation system
58     digitalWrite(BUZZER, HIGH); // Turn on the buzzer to signal very poor air quality
59   } else if (airQuality > airQualityThreshold1) {
60     digitalWrite(RELAY_VENT, HIGH); // Turn on the ventilation system
61     digitalWrite(BUZZER, LOW); // Keep the buzzer off for moderate air quality
62   } else {
63     digitalWrite(RELAY_VENT, LOW); // Turn off the ventilation system
64     digitalWrite(BUZZER, LOW); // Turn off the buzzer
65   }
66
67   // Wait for a minute before next reading
68   delay(60000);
69 }
70

```

Линк спрема спроведената симулација: <https://www.tinkercad.com/things/lmfF3LlxyX-vms-proekt>

Линк спрема спроведен комплетен проект на GitHub: <https://github.com/spishtolova/VMS-Proekt>

Заклучок

Проектот *AeroSense* успешно интегрира различни еколошки сензори и контролни механизми за одржување на оптималниот квалитет и удобност на воздухот во затворените простории. Со користење на сензор DHT22 за отчитување на температурата и влажноста и сензор MQ135 за проценка на квалитетот на воздухот, системот обезбедува сеопфатно следење на условите во околината.

Клучни Достигнувања:

Автоматска контрола на животната средина: Системот автоматски ја регулира работата на вентилаторите и грејачите врз основа на температурните прагови, обезбедувајќи удобна внатрешна клима.

Предупредувањата за вентилација и звучници: Се активираат врз основа на отчитувањата за квалитетот на воздухот, промовирајќи поздрав воздух и обезбедувајќи навремени известувања кога квалитетот на воздухот паѓа под прифатливите нивоа.

Механизам за адаптивен одговор: Со имплементирање на повеќе прагови за квалитетот на воздухот, системот нуди нијансирани одговори на различни нивоа на загадување, од умерено до сериозно.

Идни подобрувања:

Кориснички интерфејс: Податоците од сензорот во реално време континуирано се следат и се известуваат преку серискиот монитор, обезбедувајќи им на корисниците непосредни повратни информации за условите на животната средина.

Интеграција со системи за паметни домови: Потенцијална интеграција со платформи за паметни домови за да се овозможи далечинско следење и контрола.

Евиденција и анализа на податоци: Додавање карактеристики за евиденција на податоци и историска анализа за следење на долгорочните трендови и перформанси.

Напредни способности на сензорите: Вградување на дополнителни сензори за мерење на други загадувачи или параметри за посеопфатен систем за контрола на животната средина.