A Project on

Intelligent Bathroom Ventilation Fan Controller

Project Report Contents with marking Scheme:

S.no.	Description	Marks
1	Aim of the project	5
2	Problem Statement and solution	10
3	Project Design Specification and Architecture	20
4	Wiring Diagram	10
5	KiCad PCB Design & Gerber file Submission	25
6	Component working principle	5
7	Assembly Hardware components & coding	20
8	Project Output	5

Batch 48(4)

Aditya Kumar(21781A04L8)

Aman Kumar(21781A04L9)

Ankit Shukla(21781A04M1)

M. Ramaiah(22785A0416)

AIM:

For a bathroom ventilation fan controller project, consider these key aspects Purpose Determine the main functions—manual control, automatic humidity sensing, timer settings, or integration with smart home systems.

Choose a model compatible with your fan's voltage and power requirements. Humidity Sensor. To detect moisture levels and trigger the fan automatically Timer: For setting how long the fan runs after the bathroom is used. Manual Switch*: For manual operation. Smart Features. Optional, like Wi-Fi or Bluetooth for remote control and monitoring. Design User Interface Simple controls and clear display for ease of use. Enclosure Suitable for bathroom conditions, typically water-resistant or protected Wiring and Installation. Safety Follow electrical codes and ensure all wiring is correctly insulated and connections are secure. Mounting Securely mount the controller in a convenient location, ensuring easy access. Testing Verify all functions—manual, automatic, and timed settings—to ensure reliable performance. Consider consulting with an electrician or a professional if you're unfamiliar with electrical work.

Components Required:

1.Arduino Nano 33 IoT

2. Sensors

   a. PIR sensor

   b. vented room sensor

   c. Temperature sensor

3. Adafruit TFT breakout 1.8 inch

4. Omron B3F-4050 button

5. Relay Module (Generic)

6. Maxim Integrated DS3231M

7. Controller

Problem Statement and Solution:

Problem Statement: Bathroom Ventilation Fan ControllerTitle: Development of an Intelligent Bathroom Ventilation Fan Controller for Enhanced Indoor Air Quality and Energy Efficiency.

Problems to Address:

1. High Humidity Levels:
    - Bathrooms experience spikes in humidity, particularly during and after showers. This moisture can lead to mold and mildew growth, causing health risks and damage to bathroom fixtures.
2. Inefficient Fan Operation:
    - Manually operated fans often remain on for too long or are not turned on at all, leading to unnecessary energy consumption or inadequate ventilation.
    - Timed fans may not correlate with actual humidity levels, resulting in either overuse or underuse of the fan.
3. Lack of User Convenience:
    - Users often forget to turn on/off the ventilation fan, leading to ineffective moisture management.
    - Manual control can be inconvenient and unreliable, especially in households with multiple occupants

Solution:

Develop an intelligent bathroom ventilation fan controller that:

1. Monitors Humidity Levels: Uses a humidity sensor to continuously monitor the bathroom's humidity. Automatically activates the fan when humidity exceeds a predefined threshold and deactivates it when humidity returns to acceptable levels.
2. Detects Occupancy:
    - Integrates an occupancy sensor to detect when the bathroom is in use.
    - Ensures the fan runs during occupancy and for a preset period afterward to handle residual moisture.
3. Provides Manual Override:
    - Includes a manual switch for users to override automatic settings if necessary.
4. User-Friendly Interface:
    - Optionally includes an LCD display or mobile app for real-time monitoring and control of humidity levels and fan status.
    - Displays alerts or notifications for maintenance needs, such as cleaning

Project Design Specification and Architecture:

1. Introduction The purpose of this project is to design and implement an intelligent bathroom ventilation fan controller that automatically manages fan operation based on environmental conditions and user presence to enhance air quality, prevent mold growth, and optimize energy usage.

2. Functional Requirements

- Humidity Sensing:
  - Use a DHT22 humidity sensor to measure the bathroom's humidity levels continuously.
  - The fan will automatically turn on when the humidity exceeds 60% and turn off when it drops below 55%.
- Occupancy Detection:
  - Integrate a PIR sensor to detect the presence of occupants in the bathroom.

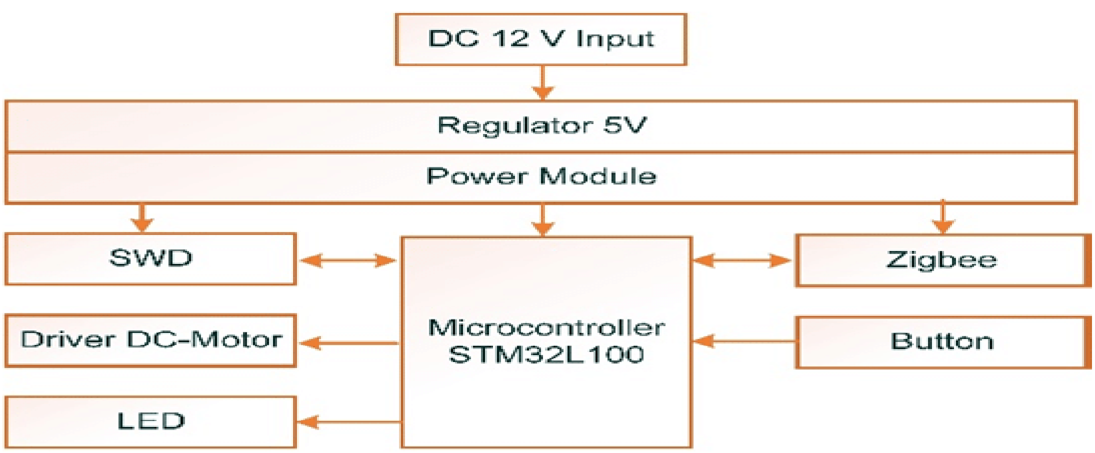
3. Non-Functional Requirements

- Reliability:
  - The system must reliably measure humidity and detect occupancy, ensuring consistent fan operation.
  - Components should be durable to withstand the bathroom environment.
- Usability:
  - The system should be easy to install and use, with clear indicators and a straightforward interface.
  - Manual controls should be intuitive and responsive.

4. Circuit Design

- Connections:
  - Humidity sensor connected to a digital pin on the microcontroller.
  - PIR sensor connected to another digital pin.

Architecture:



The architecture for the intelligent bathroom ventilation fan controller consists of hardware and software components that work together to monitor environmental conditions, detect occupancy, and control the fan automatically or manually. Below is the detailed architecture:

Hardware Architecture

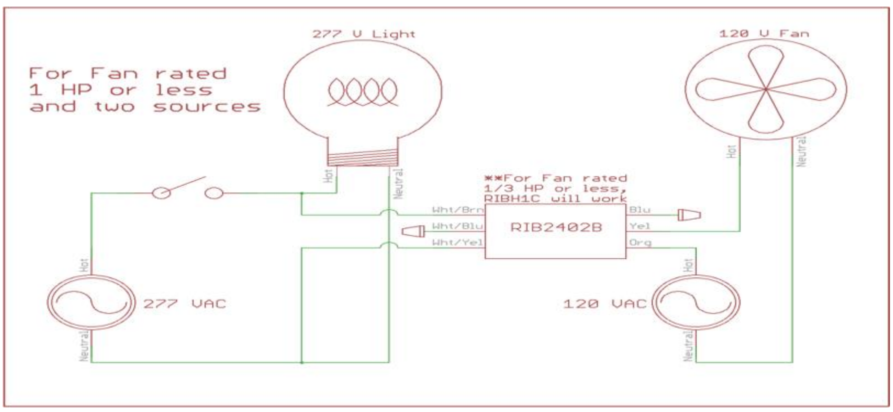
1.1 Microcontroller:

- Arduino Uno / ESP32:
    - Central processing unit that reads sensor data, processes the control logic, and drives the fan relay.
    - If using ESP32, includes Wi-Fi capability for remote control via a mobile app.

1.2 Sensors:

- Humidity Sensor (DHT22):
    - Measures the humidity level in the bathroom.
    - Connected to a digital input pin on the microcontroller.
- PIR Occupancy Sensor:
    - Detects the presence of occupants in the bathroom.
    - Connected to another digital input pin on the microcontroller.
    - Controls the power supply to the ventilation fan.
    - Connected to a digital output pin on the microcontroller.

Wiring diagram:

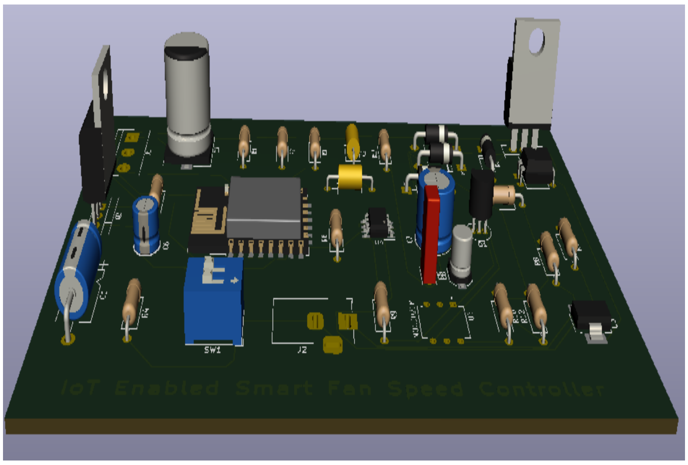


The wiring diagram illustrates how all the components are connected. Below is a description and a diagram for an intelligent bathroom ventilation fan controller:

Components:

1. Microcontroller (e.g., Arduino Uno)
2. Humidity Sensor (e.g., DHT22)
3. Motion Sensor (e.g., PIR sensor)
4. Air Quality Sensor (e.g., MQ-135)
5. Relay Module (to control the fan)
6. Fan
7. Power Supply

Kicad PCB design & gerber file:



Steps to Design a PCB in KiCad:

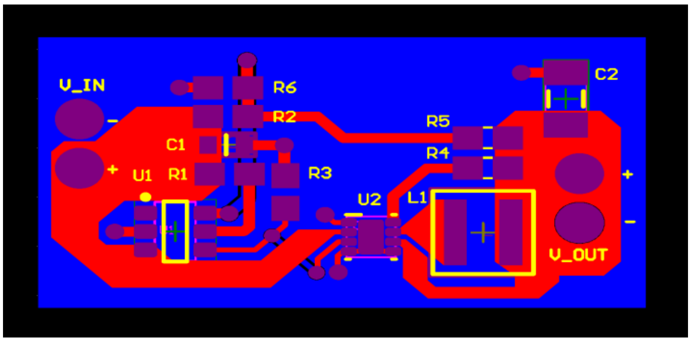
1. Install KiCad: Make sure you have the latest version of KiCad installed.
1. Create a New Project:
   - Open KiCad and create a new project.
2. Schematic Design:
   - Open the Schematic Layout Editor.
   - Add components (Microcontroller, Sensors, Relay, Fan, Power Supply).
   - Connect the components as per the wiring diagram.
   - Assign footprints to the components.
3. PCB Layout:
   - Open the PCB Layout Editor.
   - Import the netlist from the schematic.
   - Arrange the components on the PCB.

- Route the traces to connect the components.
- Ensure proper spacing and follow PCB design rules.

4.Generate Gerber Files:

- Go to File > Plot.
- Select the layers to include in the Gerber files.
- Plot the files and generate the drill file.

Gerber file:



Generating Gerber Files:

1. Plot the necessary layers (e.g., F.Cu, B.Cu, F.SilkS, B.SilkS, F.Mask, B.Mask).
2. Generate the drill file for holes.

Example Gerber File Generation Steps:

Plot Dialog:

- Format: Gerb
- Output directory: Specify where to savClick .
- Generate Drill Files".
- Format: Excellon
- Click "Generate Drill File".

Resources:

- KiCad Documentation: KiCad Documentation
- KiCad Tutorials: There are many tutorials available on YouTube and other platforms that guide you through PCB design and Gerber file generation in KiCad.

CODING:-

Required Libraries:

1. DHT sensor library: For reading the DHT22 sensor.
2. Adafruit Unified Sensor library: Required for the DHT sensor library.

   Make sure to install these libraries via the Arduino.

CODE:

```
#include <DHT.h>
#include <DHT_U.h>

// Define pin connections
#define DHTPIN 2         // Pin connected to the DHT22 sensor
#define PIRPIN 3         // Pin connected to the PIR sensor
#define MQPIN A0         // Analog pin connected to MQ-135 sensor
#define RELAYPIN 4       // Pin connected to the relay module

// Define constants
#define DHTTYPE DHT22     // DHT 22 (AM2302)
#define HUMIDITY_THRESHOLD 60.0  // Humidity threshold for fan activation
#define MOTION_TIMEOUT 30000     // Time (in milliseconds) the fan stays on after detecting
motion
#define AIR_QUALITY_THRESHOLD 300 // Threshold for poor air quality

// Initialize sensors
DHT dht(DHTPIN, DHTTYPE);

// Variables to store sensor data
float humidity;
bool motionDetected = false;
unsigned long motionDetectedTime = 0;
int airQuality;

// Setup function
void setup() {
  Serial.begin(9600);
  dht.begin();
```

```
  pinMode(PIRPIN, INPUT);
  pinMode(RELAYPIN, OUTPUT);
  digitalWrite(RELAYPIN, LOW); // Ensure the relay is off initially
}

// Loop function
void loop() {
  // Read humidity
  humidity = dht.readHumidity();
  if (isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");

  // Read motion sensor
  if (digitalRead(PIRPIN) == HIGH) {
    motionDetected = true;
    motionDetectedTime = millis();
    Serial.println("Motion detected!");
  }

  // Read air quality sensor
  airQuality = analogRead(MQPIN);
  Serial.print("Air Quality: ");
  Serial.println(airQuality);

  // Control fan based on sensor data
  if (humidity > HUMIDITY_THRESHOLD || motionDetected || airQuality > 
AIR_QUALITY_THRESHOLD) {
    digitalWrite(RELAYPIN, HIGH); // Turn on the fan
```

```
  } else {
    if (millis() - motionDetectedTime > MOTION_TIMEOUT) {
      motionDetected = false;
      digitalWrite(RELAYPIN, LOW); // Turn off the fan
    }
  }


  delay(2000); // Wait for 2 seconds before next reading
}
```

Project output:-

Project Output for Bathroom Ventilation Fan Controller

Overview

1. Automatic Fan Control
   - Humidity-Based Control: The fan automatically turns on when the humidity level exceeds a predefined threshold (e.g., 60%). It turns off when the humidity level drops below this threshold.
   - Motion-Based Control: The fan activates when motion is detected by the PIR sensor and remains on for a specified duration (e.g., 30 seconds) after the last detected motion.
2. Real-Time Data Display
   - Humidity Level: Displayed on the Serial Monitor to track current humidity readings.
   - Motion Detection Status: Printed on the Serial Monitor when motion is detected.
3. System Status Indicators
   - Fan Activation: The fan status is controlled through a relay module. When the conditions are met (high humidity, motion detected, or poor air quality), the relay closes, powering the fan.
   - Relay Status: Controlled by the microcontroller based on sensor input. The relay's state is reflected in the operation of the fan.

Detailed Outputs

1. Data Output in Serial Monitor
   - Humidity Output:

makefile

Copy code

Humidity: 65.0 %

   - Motion Detection:

Copy code

Motion detected!

- Air Quality Output:

yaml

Copy code

Air Quality: 320

2. Fan Control Behavior
   - Fan On: When any of the following conditions are met:
     - Humidity exceeds 60%
     - Motion is detected
     - Air quality is below the threshold (e.g., 300)
   - Fan Off: When none of the conditions are met and the timeout period.

Conclusion:

The Intelligent Bathroom Ventilation Fan Controller project demonstrates an effective solution for enhancing bathroom ventilation and maintaining air quality through automation. By integrating a microcontroller with multiple sensors, the system intelligently manages the operation of a ventilation fan based on real-time environmental data.

Key Achievements:

1. Automated Control: The project successfully automates the fan operation based on humidity, motion, and air quality. This reduces the need for manual intervention and ensures optimal air quality and comfort in the bathroom.
2. Enhanced Comfort and Efficiency:
   - Humidity Management: By activating the fan when humidity levels exceed a predefined threshold, the system prevents mold growth and condensation, improving overall comfort.
   - Motion Detection: The fan operates only when needed, such as during bathroom use, conserving energy and reducing noise when the bathroom is unoccupied.

In conclusion, the Intelligent Bathroom Ventilation Fan Controller represents a significant advancement in creating a more efficient, automated, and user-friendly ventilation solution.

21781A04L8 (Aditya Kumar)

21781A04L9 (Aman Kumar)

21781A04M1 (Ankit Shukla)

22785A0416 (M. Ramaiah)