

**Department of Computer Science and Engineering**

| **Course Code:** CSE461 | **Credits:** 1.5 |
| --- | --- |
| **Course Name:** Introduction to Robotics Lab | **Semester:** Spring2024 |

**Faculty:** Shakir Rouf  **Faculty:** Niloy Irtisam

**Group 07**

**Introduction to Robotics: CSE461 - Hardware Project**

**Smart Home Automation System with Sensor-Driven Environmental Control and Safety Feature**

**Group Members:**

**Name:** MD Ikramul Kayes  **ID:**21301576

**Name:** Umme Abira Azmary **ID:**20101539

**Name:** Fareha Alam **ID:**21301391

**Name:** Noshin Anjum Malik Piasha **ID:** 21201340

**Name:** Umma Faria Rahman  **ID:**20301346

**Objective:**

Design and implement a comprehensive smart home automation system that utilizes sensor-driven environmental monitoring and control as well as incorporate safety features to enhance the comfort, convenience, and security of occupants.

**Project Description:**

The Smart Home Automation System is a cutting-edge solution designed to provide sensor-driven environmental control to enhance the comfort, convenience, and safety of residential environments through intelligent automation. It integrates a range of components, including an Arduino Nano microcontroller, various functional sensors, actuators, and supporting modules. Here, we have used the Arduino Nano microcontroller as the brain of the system to coordinate interactions between sensors, actuators, and the user interfaces. The working functionality of sensors is described below:

**Ultrasonic Sensor:** Upon detecting the presence of a person entering a room, the ultrasonic sensor communicates with the Arduino controller and activates the fan through a relay module. This functionality automates the connection and optimizes energy usage by activating cooling only when necessary, based on occupancy.

**Temperature Sensor:** This sensor has the capability to detect the temperature of the environment. Upon detection, the system dynamically adjusts the fan speed in response to changes in room temperature. By continuously monitoring temperature levels, the system ensures optimal thermal comfort while minimizing energy consumption.

**Natural Gas Detector Sensor:** The natural gas detector sensor detects the presence of unwanted smoke or gas leak from pipelines, connections, or appliances. In the event of a gas leak, the sensor disconnected the drive module connection to cut off the power supply to prevent potential hazards.

**Rain Detector Sensor:** To protect the room from untimely rain or accidental water spillage, the system features a rain detection sensor. Upon detecting moisture, this sensor activates a Servo motor, automatically closing windows to prevent water ingress.

**Equipements:**

1. Arduino Nano
2. Ultrasonic sensor.
3. Dht11
4. Rain sensor
5. Motor driver
6. Relay module
7. Cooling fan
8. Battery
9. Lm2596
10. Buzzer
11. SG90 servo motor
12. Jumper wire
13. Gauge wire
14. Breadboard
15. Rockar switch
16. MQ2 Gas sensor
17. Lcd 1602 with i2c

**Circuit Diagram/Schema of the project:**

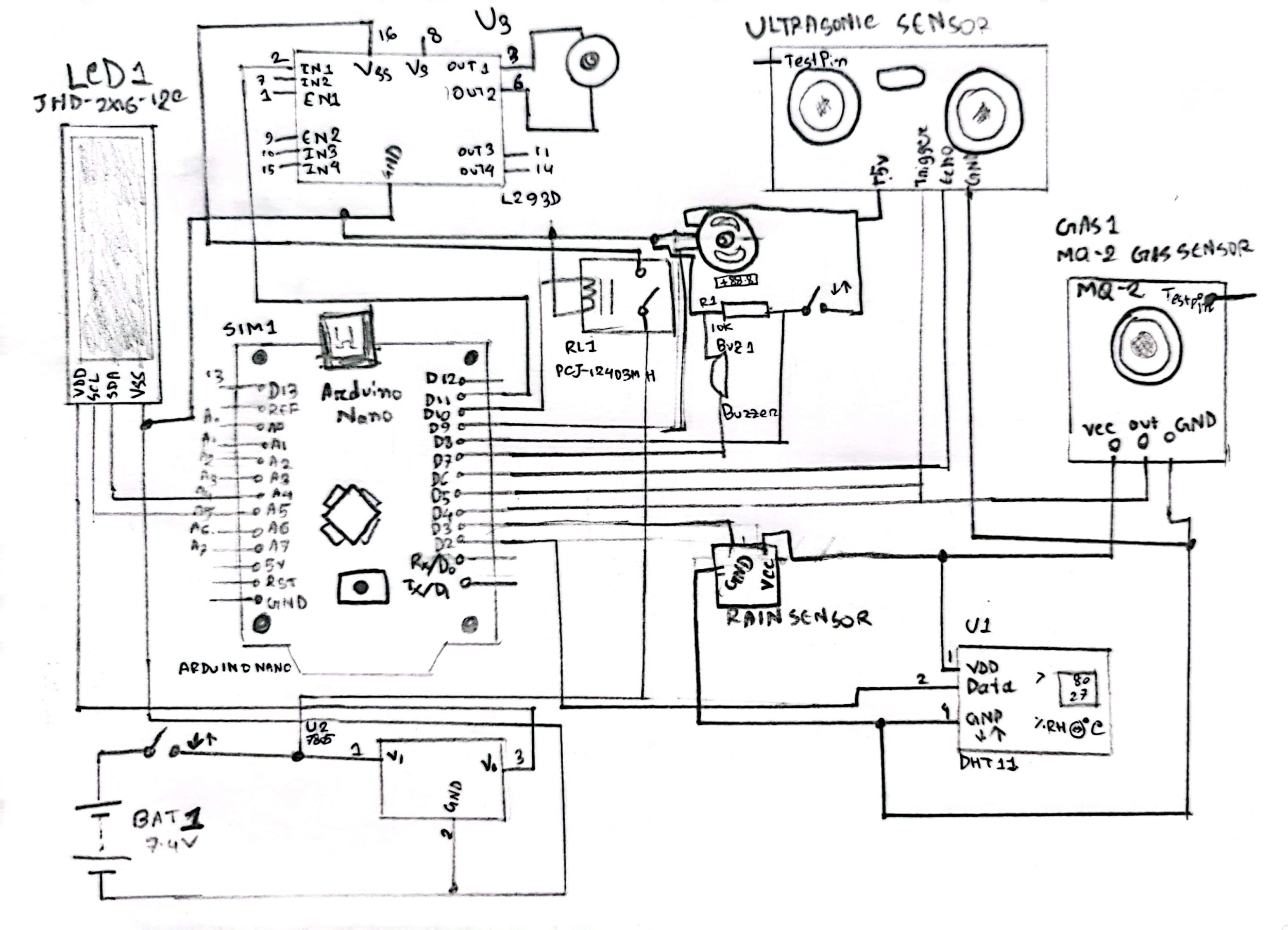
****

Figure 1: circuit diagram of the experiment

**Experimental Setup:**

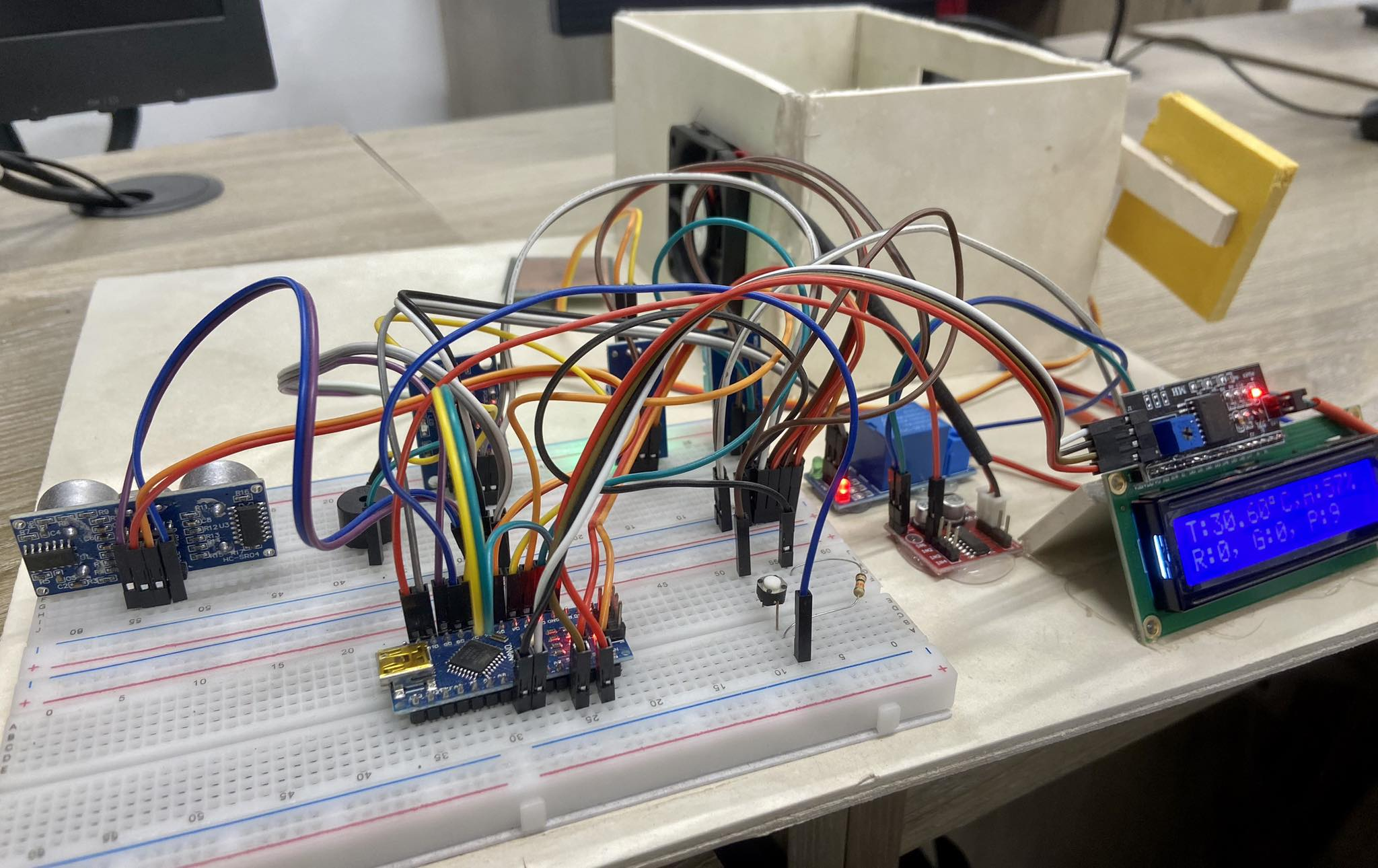
****

Figure 2: implementation of the experiment

We have used a 7.5V battery to supply power in this project. The power distribution is described below:

(i) The 7.5V is converted to 5V through the Bug Module and supplied to all the sensors, Arduino Nano and LCD screen.

(ii) The cooling fan, which we are using in the room’s fan, functions at 7.5V. So, we are applying 7.5V to the motor driver through the Relay Module. The relay module works as a switch which is controlled by the Arduino.

Next, the Arduino Nano microcontroller coordinates interactions between sensors, actuators, and user interfaces. To elaborate, the temperature sensor is connected to pin 2, the Rain sensor to pin 3, the Gas sensor(MQ2) to pin 4, the Ultrasonic sensor’s trigger to pin 5 and echo to pin 6, the Buzzer to pin 7, the fan button to pin 8, Relay module to pin 10 and Fan to pin 11 of the Arduino Nano. As a result, the data pins of the sensors complete their functionalities through the Arduino.

We directly connected power from the battery to relay modules input and output terminal of the relay is connected to the motor driver. When the Arduino passes high in the data pin of Relay only then the Fan starts moving. So, when the MQ2 gas sensor detects gas, it sends high output to the Arduino pin, only then Arduino send a Low value to Relay which cuts off the power supply to motor driver and vice versa. The ultrasonic sensor detects a human if the human stands in front of it. Ultrasonic’s transmitter transmits and the receiver receives the signal. Based on transmitting and receiving signal the Arduino calculates distance. If the distance is less than or equal to 5 meter then it starts the fan if more than one person is inside a room. Moreover, if any rain drops drop on top of the rain sensor it passes high input to Arduino. Upon receiving the signal Arduino nano sets the value of servo to 180, so the window gets closed. If no rain gets detected by rain sensor then it sets the servo to 0 degree. Furtheremore, we have a temperature sensor, which senses temperature and sends it to Arduino. Based on the temperature, the Arduino sets the pwm of the motor driver and sends the pwm to the motor driver. The motor driver then speeds ups the fan if the pwm is high or speeds down the fan if the pwm value is low.

**User Manual:**

Users must connect the sensors and equipment with the Arduino Nano to turn on the automation system.

For the ultrasonic sensor, users need to connect the Vcc, Ground and then the echo and trigger pin with the Arduino Nano. After successful connection, when a user enters the room, the ultrasonic sensor automatically detects the presence of that person and turns on the fan.

After this, user need to connect the Vcc, Ground from the voltage source and data pin of the temperature sensor with the Arduino Nano so that if any unwanted smoke or CH4 gas leaks in the room, the buzzer turns on and turns off the power supply to let the residents know about the danger.

Also, the user needs to set the temperature sensor with Arduino Nano. If the temperature of the room increases/decreases, the fan automatically increases/decreases the speed of the fan.

Moreover, if any unwanted water/rain splashes into the room, the connection of the rain sensor with the help of the servo motor, the window of the house will automatically turn off.

After one time setup, the program will function automatically so the user doesn't have to set up manually every time.

**Code:**

#include "DHT.h"

#include <LiquidCrystal\_I2C.h> //Importing libraries

#include <Servo.h>

Servo myservo; //initializing

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// Setting the pin of Nano

#define DHTPIN 2

#define RAIN 3

#define MQ2 4

#define trigPinA1 5

#define echoPinA1 6

#define BUZZER 7

#define BOTTOM 8

#define RELAY 10

#define FAN 11

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

int count = 0;

void setup() {

lcd.init();

lcd.backlight();

//setting the input and output pin

Serial.begin(9600);

pinMode(RAIN, INPUT);

pinMode(MQ2, INPUT);

pinMode(trigPinA1, OUTPUT);

pinMode(echoPinA1, INPUT);

pinMode(BUZZER, OUTPUT);

pinMode(BOTTOM, INPUT);

pinMode(RELAY, OUTPUT);

pinMode(FAN, OUTPUT);

myservo.attach(9);

myservo.write(0);

digitalWrite(BUZZER, LOW);

digitalWrite(RELAY, HIGH);

dht.begin();

}

void loop() {

//reading temperature and humidity

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h) || isnan(t)) {

t = 0;

h = 0;

}

int RAINState = !digitalRead(RAIN); //getting rain state

int MQ2State = !digitalRead(MQ2); //getting the gas sensor state

int dis = sennorA1(); // reading distance

int BOTTOMState = digitalRead(BOTTOM); // reading button state

if (dis <= 5 && count < 10) { //increasing the number of person

count++;

if (count >= 10) {

count = 10;

}

delay(500);

}

if (BOTTOMState == 1 && count > 0) {

count = 0;

delay(500);

}

if (RAINState == 1) { // if detects rain

myservo.write(180);

}

else {

myservo.write(0);

}

if (MQ2State == 1) { // if detects gas

digitalWrite(RELAY, LOW);

digitalWrite(BUZZER, HIGH);

}

else {

digitalWrite(RELAY, HIGH); // if do not detect gas

digitalWrite(BUZZER, LOW);

}

int pwm = map(int(t), 30, 45, 50, 255); // mapping pwm based on temp

if (pwm <= 50) { // if pwm is too low

pwm = 50;

}

if (pwm >= 255) { //if pwm is too high

pwm = 255;

}

if (count > 0) { //if any body enters the room

analogWrite(FAN, pwm);

}

else {

analogWrite(FAN, 0);

}

Serial.print(t);

Serial.print("\t");

Serial.print(h, 0);

Serial.print("\t");

Serial.print(RAINState);

Serial.print("\t");

Serial.print(MQ2State);

Serial.print("\t");

Serial.print(dis);

Serial.print("\t");

Serial.print(BOTTOMState);

Serial.print("\t");

Serial.print(count);

Serial.print("\t");

Serial.print(pwm);

Serial.println();

//led display

lcd.setCursor(0, 0);

lcd.print("T:");

lcd.print(t);

lcd.print((char)223);

lcd.print("C,H:");

lcd.print(h, 0);

lcd.print("% ");

lcd.setCursor(0, 1);

lcd.print("R:");

lcd.print(RAINState);

lcd.print(", G:");

lcd.print(MQ2State);

lcd.print(", P:");

lcd.print(count);

lcd.print(" ");

delay(1);

}

// detecting distance by ultrasonic sensor

long sennorA1() {

digitalWrite(trigPinA1, LOW);

delayMicroseconds(2);

digitalWrite(trigPinA1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPinA1, LOW);

long duration = pulseIn(echoPinA1, HIGH);

int distance = duration \* 0.034 / 2;

return distance;

}

**Code Explanation:**

In the beginning we import libraries like DHT.h, LidquidCrystal\_I2C.h and Servo.h for our DHT11 sensor, LCD display and Servo motor. After importing we initialize these libraries Servo myservo and LiquidCrystal\_I2C lcd(0x27,16,2);

After this we defined the pin of the Arduino Nano in some specific variables. We also initialized the variable count = 0 for later use

In the void setup we initialized the LCD and set the pinmode as INPUT or OUTPUT. We also initialized initial states of the Servo to 0 degrees, the Buzzer to Low and the Relay to High. We even start taking inputs by starting the DHT sensor.

In the void loop we have taken humidity and temperature to h and t variables. We even set the condition where if any of the variables got Nan value it will set the variable values to 0.

We also took the value of Rain sensor to RAINState variable also took the MQ2 gas sensor value to MQ2State variable. We also have a sensorA1 function where we take the ultrasonic sensor data and in return, get the distance. So we are reading the distance in dis variable. We are also reading the state of the button in BOTTOMState variable.

In the following function, If the dis variable is less than or equal to 5 and the count is less then 10 only then we are increasing the value of the count. We are also delaying 500 so that the Arduino do not get overflowed with data. This condition increases the number of person by increasing the count variable’s value if the ultrasonic sensor detects a person. We are even checking if the BOTTOMState is 1(HIGH) and the count is greater than 0 then we are setting the count = 0. By this only by pressing a button, we can reset the person count number.

If the RAINSTATE is 1(HIGH) then we are setting the myservo to 180 degree which closes the window. Otherwise, we are setting the myservo to 0, which makes the window is open. If the MQ2State is 1 then we are setting the state of RELAY to Low and Buzzer to High. Otherwise, we are setting the RELAY to HIGH and BUZZER to Low. By this if the Gas sensor detects gas(CH4 or Smoke) it turns off the RELAY by this the motor driver power gets cut off and current gets flown through the buzzer. By this Buzzer makes sound. The opposite happens when the gas sensor detects no gas.

We are mapping the pwm where we took the threshold from 30 degree to 45 degree which gets converted to 50 to 255 pwm value. Based on the temperature the value of pwm is getting set up. If the temparature less or equal to 30 then we are setting the pwm value to 50. If the temparature value is greater than 45 then we are setting the value of pwm to 255. We have some bunch of conditions like if the count is greater than 0 then we are setting the pwm of FAN to pwm value we initialized in pwm variable before. Otherwise, if the condition is not satisfied we are setting the pwm value of the FAN to 0.

After that we are writing the Serial for debugging purposes. We even printed the value of temperature and humidity from t and h variables. We are even writing the rain state, gas state and count of persons in the lcd display.

**Troubleshoot:**

1. If any of the sensors do not work just change the position of the column the sensor is placed in on the breadboard. If it do not work then change the wires.
2. Sometimes the Gas sensor does not work properly, as it holds the gas in its holes. So provide as much Gas as you can to test the system.

**Video Demonstration:**

<https://youtu.be/7NGtZsp_PlA>