**Project 2- Smart Home Monitoring System using Arduino**

The project is a **Smart Home Monitoring System** that measures temperature, humidity, and proximity of objects, and reacts with alarms, displays information on an LCD, and controls a servo motor.

**Components:**

* Arduino Uno
* DHT22 (Temperature & Humidity Sensor)
* Ultrasonic sensor (HC-SR04)
* LED
* Buzzer
* Servo motor
* 16x2 LCD (with I2C adapter for easier connection)
* Jumper wires
* Resistors (220Ω for LED)
* Breadboard
* Potentiometer (for LCD contrast adjustment)
* Power supply (via USB or battery)

**Concept:**

1. The DHT22 sensor will monitor the room's temperature and humidity.
2. The ultrasonic sensor will detect proximity to alert when an object or person is too close.
3. The LCD will display real-time temperature, humidity, and distance readings.
4. If an object comes within a defined range (e.g., 20 cm), the buzzer will sound, the LED will blink, and the servo motor will move.
5. The system provides a visual and audible warning when someone is too close, making it ideal for use as a smart door alert system.

**Circuit Connections:**

1. **DHT22 Sensor**
   * VCC → 5V
   * GND → GND
   * Data → Digital Pin 2
2. **Ultrasonic Sensor (HC-SR04)**
   * VCC → 5V
   * GND → GND
   * Trig → Digital Pin 9
   * Echo → Digital Pin 8
3. **LCD with I2C Adapter**
   * VCC → 5V
   * GND → GND
   * SDA → A4
   * SCL → A5
4. **LED**
   * Anode (+) → Digital Pin 10 (with 220Ω resistor)
   * Cathode (-) → GND
5. **Servo Motor**
   * Signal pin → Digital Pin 11
   * Power pin → 5V
   * Ground pin → GND
6. **Buzzer**
   * Positive pin → Digital Pin 12
   * Negative pin → GND

**Arduino Code:**

This code combines all the sensors and actuators. The ultrasonic sensor measures distance, the DHT22 measures temperature and humidity, and the LCD displays these values. When an object comes too close, the LED blinks, the buzzer sounds, and the servo motor moves.

#include <DHT.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Servo.h>

// Define DHT sensor

#define DHTPIN 2

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// Define ultrasonic sensor pins

const int trigPin = 9;

const int echoPin = 8;

// Define pins for LED, buzzer, and servo

const int ledPin = 10;

const int buzzerPin = 12;

Servo myServo;

// Create LCD object with I2C address 0x27

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

// Variables for ultrasonic sensor

long duration;

int distance;

void setup() {

// Initialize serial communication

Serial.begin(9600);

// Initialize DHT sensor

dht.begin();

// Initialize LCD

lcd.init();

lcd.backlight();

// Initialize pins

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(ledPin, OUTPUT);

pinMode(buzzerPin, OUTPUT);

// Attach servo to pin

myServo.attach(11);

myServo.write(0); // Set servo to 0 degrees initially

// Print startup message on LCD

lcd.setCursor(0, 0);

lcd.print("Smart Home Sys");

lcd.setCursor(0, 1);

lcd.print("Initializing...");

delay(2000); // Wait 2 seconds

lcd.clear();

}

void loop() {

// Get temperature and humidity from DHT22 sensor

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

// Check if readings are valid

if (isnan(temperature) || isnan(humidity)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

// Get distance from ultrasonic sensor

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

// Print sensor values to Serial Monitor

Serial.print("Temp: ");

Serial.print(temperature);

Serial.print(" C, Humidity: ");

Serial.print(humidity);

Serial.print("%, Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Display data on LCD

lcd.setCursor(0, 0);

lcd.print("Temp: ");

lcd.print(temperature);

lcd.print("C");

lcd.setCursor(0, 1);

lcd.print("Hum: ");

lcd.print(humidity);

lcd.print("% Dist: ");

lcd.print(distance);

lcd.print("cm");

// Control based on distance

if (distance < 20) {

// If object is close, trigger alerts

digitalWrite(ledPin, HIGH);

tone(buzzerPin, 1000); // Buzzer sounds

myServo.write(90); // Servo moves to 90 degrees

delay(1000); // Keep the servo in this position for 1 second

myServo.write(0); // Reset servo to 0 degrees

} else {

// If object is far, turn off LED and buzzer

digitalWrite(ledPin, LOW);

noTone(buzzerPin);

}

delay(2000); // Delay before next loop

}

**How it Works:**

1. **DHT22 Sensor**: The sensor continuously monitors temperature and humidity. This information is shown on the LCD and sent to the Serial Monitor for tracking.
2. **Ultrasonic Sensor**: The HC-SR04 sensor detects the distance to an object. If the object is closer than 20 cm, it triggers an alarm.
3. **LCD Display**: The LCD provides real-time feedback, showing temperature, humidity, and the distance to an object.
4. **Servo Motor**: The servo motor reacts when the object is too close (less than 20 cm). It rotates to 90 degrees and returns to 0 degrees after a short delay.
5. **Buzzer and LED**: If an object is detected within the danger zone, the buzzer will sound, and the LED will blink, providing visual and audible feedback.

**Additional Features:**

* **LCD Feedback**: The LCD constantly updates, showing the current room temperature, humidity, and proximity, making it useful for a variety of monitoring applications.
* **Object Detection**: The ultrasonic sensor allows the system to detect when someone approaches, which could be used for automatic door control or security alerts.
* **Environmental Monitoring**: The DHT22 provides useful environmental data, allowing the system to alert users to temperature or humidity changes.

**Applications:**

* **Smart Doorbell System**: Automatically detect when someone is at the door and provide visual/audible feedback.
* **Home Automation**: Control room appliances based on proximity, temperature, or humidity.
* **Security System**: Detect movement near doors or windows and trigger alarms.

This project is a versatile home monitoring system that can be further extended with features like wireless connectivity or integration with a mobile app for remote monitoring.