Smart Home Automation System using Arduino

- **Project Report -**

F H D N SILVA

AS2020931

MAT/PHY/ICT

Acknowledgment

While working on this project I have received invaluable help from many people. I would like to take this opportunity to express my deepest gratitude to everyone who helped me. I felt grateful to express my boundless honor and respect to Mr. M.D.R. Perera Senior Lecture, Department of Computer Science, Faculty of Applied Science, University of Sri Jayewardenepura, for his deep knowledge that influenced me to carry out this project. Also, grad to my family members and other faculty members for their kind help to finish my project.

Contents

Smart Home Automation System using Arduino

1. Introduction:

Electricity expenses in Sri Lanka have soared, prompting a need for innovative solutions to curb costs and promote energy efficiency. In response to this challenge, the Smart Home Automation System was conceptualized as a strategic approach to automating household devices and optimizing power consumption.

The central focus of this project lies in a practical demonstration involving the automation of a room's fan. Recognizing the potential impact of air circulation on energy consumption, the system is designed to activate the fan automatically when the room temperature surpasses a predefined threshold. Once the temperature returns to the normal range, the fan is programmed to deactivate, ensuring energy is utilized only when necessary.

2. Features:

In the pursuit of an energy-efficient household, the Smart Home Automation System encompasses a range of key features to deliver a comprehensive solution:

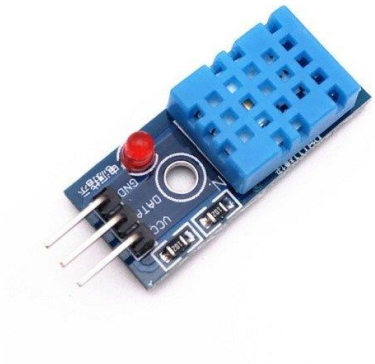
- **Automatic Fan Activation:** The core functionality of the system involves the automatic activation of the fan in response to temperature fluctuations, ensuring optimal air circulation for comfort and energy efficiency.
- **SMS Notifications:** Users are kept informed about the system's operation through SMS notifications. When the fan turns on or off, timely messages are sent to the user, providing real-time updates and fostering an interactive user experience.
- **Voice Control:** Enhancing user convenience, the system incorporates voice control functionality. Users can effortlessly command the fan's operation through voice commands, offering a hands-free and intuitive control mechanism.
- **Third-Party Mobile App Control:** The inclusion of a third-party mobile app extends control beyond the physical confines of the home. Users can remotely manage the fan, adjusting settings and monitoring its status from the convenience of their smartphones.

3. Used Components:

The system's architecture is built upon a selection of carefully chosen components, each playing a vital role in ensuring seamless operation and efficient energy utilization:

- **Arduino Uno Board:** Serving as the project's central processing unit, the Arduino Uno board orchestrates communication between various components and executes programmed functionalities.

- **DHT11 Humidity and Temperature Sensor:** This sensor is integral for the accurate measurement of room temperature, enabling the system to respond dynamically to environmental conditions.



- **HCO6 Module:** Playing a pivotal role in facilitating communication, the HCO6 module ensures a reliable connection between the Arduino board and the third-party mobile app.

HC-06

- **GSM Module:** Responsible for sending SMS notifications, the GSM module enhances user awareness by providing instant updates on the fan's status.

- **LED Bulbs:**

- **12V Cooling Fan:** The target of automation, the 12V cooling fan is activated and deactivated based on the system's programmed conditions, contributing to energy-efficient air circulation.

- **Single Potentiometer:** This component enables users to set the temperature threshold for automatic fan activation manually.

4. Method:

The operational methodology of the Smart Home Automation System revolves around a carefully orchestrated interplay of hardware and software components. While the primary focus is the automation of the fan, the system offers multiple control avenues for users:

- **Third-Party Mobile App:** Although a dedicated mobile app wasn't developed for this iteration, the system leverages a third-party mobile app for user interaction. Through this app, users can manually control the fan, set temperature thresholds, and receive notifications.
- **Voice Control:** Users have the option to control the fan using voice commands, providing a hands-free alternative for seamless integration into daily routines.
- **Potentiometer for Manual Control:** The single potentiometer allows users to manually set the temperature threshold, providing a tactile and straightforward means of interaction.

5. System Overview:

The Smart Home Automation System operates on a user-centric model, ensuring ease of use and practicality. Users have the flexibility to set temperature thresholds either manually through the potentiometer or via the third-party mobile app. The system responds dynamically to temperature changes, activating or deactivating the fan accordingly. SMS notifications serve as an additional layer of user engagement, providing instant updates on the fan's status.

While the current version lacks specific security measures, the system's interaction is confined to the user's local environment, minimizing potential security risks associated with broader network connectivity.

6. Structure of the Prototype:

The physical manifestation of the Smart Home Automation System involves an intricately connected network of components:

- **Arduino Uno Board:** Positioned as the brain of the operation, the Arduino Uno board serves as the central hub, receiving inputs from sensors, processing data, and executing commands.
- **DHT11 Humidity and Temperature Sensor:** Strategically placed to capture accurate room temperature data, the DHT11 sensor plays a crucial role in the system's ability to make informed decisions about fan activation.
- **CO6 Module:** Ensuring stable and reliable communication, the CO6 module facilitates the interaction between the Arduino board and the third-party mobile app.
- **GSM Module:** Positioned for optimal connectivity, the GSM module enables the system to send SMS notifications to the user, keeping them informed about the fan's status.
- **12V Cooling Fan:** The focal point of automation, the 12V cooling fan is activated or deactivated based on the user-defined temperature threshold, contributing to a controlled and energy-efficient environment.
- **Single Potentiometer:** Placed for user accessibility, the potentiometer allows manual adjustment of the temperature threshold, providing users with a tangible means of customization.

The absence of a cloud service indicates a deliberate choice to keep the system's operations localized and avoid potential external vulnerabilities. Instead, the reliance on a third-party mobile app streamlines user interactions and data management.

7. Issues:

Despite the system's functionality, a significant limitation is its confinement to a single room. This constraint arises from the use of an Arduino Uno board, limiting the range of communication and control to a Bluetooth radius. As a result, the system's impact is localized, offering room-specific automation.

8. Limitations:

The system's reach is inherently limited due to its reliance on the Arduino Uno board and Bluetooth connectivity. Consequently, the range of control and automation is confined to the proximity of the user within the Bluetooth range, restricting its application to a single room.

9. Conclusion:

In conclusion, the Smart Home Automation System represents a proactive step towards addressing the escalating costs of electricity in Sri Lanka. By focusing on a practical demonstration involving fan automation, the system showcases the potential for energy savings and enhanced user convenience.

10. Code Overview:

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

SoftwareSerial sim900A(7, 8);  // RX, TX pins on Arduino

#define DHTPIN 2    // Pin where the DHT11 is connected
#define DHTTYPE DHT11   // DHT 11
#define Fan 4

bool state = true;

DHT dht(DHTPIN, DHTTYPE);
const int potPin = A0;  // Pin connected to the potentiometer

void setup() {
  Serial.begin(9600);

  sim900A.begin(9600);

  delay(2000);
  dht.begin();
  pinMode(Fan,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
}
```

```
void loop() {
  int potValue = analogRead(potPin);  // Read the analog value from the potentiometer
  int val = map(potValue,0,1023,0,40);

  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  else if(temperature > val){
    Serial.print("\tFan Is On...!");
    digitalWrite(Fan,HIGH);
    if(state == true){
      sendSMS("+94725815118", "Temprature is Grather than Set Value.Fan is On!");  // Replace with the recipient's phone number
      state = false;
    }
  }else{
    digitalWrite(Fan,LOW);
    state = true;
  }

  Serial.print("Set Temprature Value: ");
  Serial.print(val);

  Serial.print("\t Temperature: ");
  Serial.println(temperature);

  delay(100);  // Add a delay to make the readings more readable
```

```
  if (Serial.available() > 0) {
    char data = Serial.read(); // Read the incoming data
    Serial.print("Received: ");
    Serial.println(data);

    if (data == '1'){
      digitalWrite(9,HIGH);
    }
    if(data == '2'){
      digitalWrite(9,LOW);
    }
    if (data == '5'){
      digitalWrite(11,HIGH);
    }
    if(data == '6'){
      digitalWrite(11,LOW);
    }
  }

}

void sendSMS(String phoneNumber, String message) {
  sim900A.println("AT+CMGF=1");  // Set SMS mode to text
  delay(1000);

  sim900A.print("AT+CMGS=\"");
  sim900A.print(phoneNumber);
  sim900A.println("\"");

  delay(1000);

  sim900A.print(message);
  delay(100);
  sim900A.write(26);  // Send Ctrl+Z to indicate the end of the message
  delay(1000);

  Serial.println("SMS Sent!");

}
```

14. Budget:

Arduino UNO Original Development Board with USB Cable (DB0002)	Rs.2950.00
DHT11 Temperature and Relative Humidity Sensor	Rs. 290.00
Wirewound Precision Potentiometer	Rs. 520.00
Cooling Fan 12V	Rs.1050.00
Bluetooth Serial Adapter Board for HC05 HC06	Rs. 150.00
SIM900A GSM GPRS Module with Antenna	Rs.1900.00
Jumping Wires	Rs. 320.00
Total	Rs.7180.00

13. Future Implementation:

Recognizing the system's current limitations, future implementations present avenues for expansion and refinement:

- **Enhanced Security Measures:** As the system evolves, implementing robust security measures, such as encryption protocols and authentication methods, will be crucial for safeguarding user interactions and data.
- **Scalability with NodeMCU Board:** Transitioning to a NodeMCU board holds the promise of broader connectivity and scalability. This evolution would enable the system to extend its reach beyond a single room, transforming it into a comprehensive and versatile home automation solution.
- **Diversification of Controlled Devices:** The integration of additional home devices, beyond the fan, opens opportunities for a more holistic and tailored smart home experience. Automating appliances such as bulbs, fridges, rice cookers, ovens, and more broadens the system's utility and impact.

As the Smart Home Automation System evolves, these future implementations promise to elevate its capabilities, providing users with an enhanced and more expansive home automation experience.