Smart Home – Internet of Things Project

This project implements a smart home automation system using an Arduino microcontroller, various sensors, and actuators. The system is designed to control climate, lighting, and security, and it provides remote monitoring and control through ThingSpeak and email.

Project Objectives

- **Climate Control:** Maintain a comfortable room temperature by automatically controlling heating and cooling systems.
- **Lighting Control:** Optimize lighting based on ambient illumination and motion detection.
- **Remote Monitoring and Control:** Monitor and control the system remotely via ThingSpeak and email.
- **Data Logging and Reporting:** Generate daily reports with historical data and graphs.

System Architecture

The system consists of the following components:

Hardware:

- Arduino Uno microcontroller
- DHT11 temperature and humidity sensor
- LM35 temperature sensor
- Photoresistor
- HC-SR501 PIR motion sensor
- Relays for controlling heater, fan, and LED light

- Heater (resistor)
- DC motor (fan)
- White LED diode
- Ethernet Shield or Wi-Fi module for internet connectivity

Software:

- Arduino IDE for programming the Arduino microcontroller
- Python script for data logging, remote control, and reporting (JetBrains PyCharm)
- ThingSpeak platform for data visualization and monitoring
- Email service for notifications and remote control

System Design and Implementation

Climate Control:

- The LM35 temperature sensor measures the room temperature.
- If the temperature falls below 17°C, the heater is turned on via a relay.
- If the temperature rises above 23°C, the fan is turned on via a relay.
- The heating and cooling systems are mutually exclusive, ensuring they don't operate simultaneously.
- The climate control system can be remotely enabled or disabled via email.

Lighting Control:

- The photoresistor measures ambient illumination.
- In "auto mode," the LED light is turned on if the illumination falls below 30% and off otherwise.

- Manual light control (on/off) is available through serial communication with a PC, which automatically deactivates "auto mode."

Remote Monitoring and Control:

- Temperature, humidity, and illumination measurements are sent to ThingSpeak every 10 minutes.
- The Python script reads emails for remote control commands (e.g., turning on/off lights, climate control, or modes).
- Email notifications are sent for motion detection.

Data Logging and Reporting:

- The Python script collects daily data, including minimum, maximum, and average temperature and illumination, as well as motion detection counts.
- A daily report is generated, including graphs of temperature and illumination measurements and motion detection counts, and sent via email.

Python Script (script.py file)

The Python script handles the following tasks:

- Reads data from the Arduino's serial port.
- Processes the received data and sends it to ThingSpeak.
- Checks for email commands and sends commands to the Arduino accordingly.
- Collects daily data and generates a report with graphs.
- Sends the daily report via email.

Project Setup and Configuration

1. **Hardware Setup:** Connect the sensors, actuators, and relays to the Arduino according to the circuit diagram.
2. **Software Setup:** Install the necessary libraries (DHT, ThingSpeak) in the Arduino IDE.
3. **Arduino Code:** Upload the Arduino code to the microcontroller.
4. **Python Script:** Configure the Python script with your email credentials, ThingSpeak API keys, and serial port settings.
5. **Run the Python Script:** Execute the Python script to start data logging, remote control, and reporting.