Group no. 8

Project Topic: **Intelligent Home Automation System Using Gen AI and IoT for Personalized Energy Management to Reduce Carbon Footprint**

To set up your Intelligent Home Automation System using ThingSpeak, Arduino IDE, and IoT for personalized energy management, follow these steps:

Requirements

1. Hardware:

- Arduino board (e.g., Arduino Uno, Mega)
- ESP8266 or ESP32 module for Wi-Fi connectivity
- Sensors (e.g., temperature, humidity, light, motion)
- Actuators (e.g., relays, smart plugs)
- Jumper wires and breadboard

2. Software:

- Arduino IDE
- ThingSpeak account
- MATLAB license (for advanced data processing on ThingSpeak)

Steps

1. Set Up ThingSpeak

1. Create a ThingSpeak Account:

- Go to [ThingSpeak](https://thingspeak.com/).
- Sign up and create a new account.

2. Create a New Channel:

- Log in to your ThingSpeak account.
- Navigate to "Channels" and click "New Channel."
- Fill in the necessary details (e.g., channel name, field labels for the data you will collect such as temperature, humidity).
- Click "Save Channel."

3. API Key:

- Go to the "API Keys" tab of your channel.
- Note the "Write API Key" and "Read API Key" which you will use in your Arduino code.

2. Set Up Arduino IDE

1. Install Arduino IDE:

- Download and install the Arduino IDE from [Arduino's official website](https://www.arduino.cc/en/software).

2. Install Board Manager:

- Open Arduino IDE.
- Go to File > Preferences.
- In the "Additional Board Manager URLs" field, add:

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
http://dl.espressif.com/dl/package_esp32_index.json
```

- Click "OK."

3. Install ESP8266/ESP32 Boards:

- Go to Tools > Board > Boards Manager.
- Search for ESP8266 and ESP32, and install the packages.

3. Connect Sensors and Actuators

1. Wiring:

- Connect your sensors and actuators to the Arduino/ESP board following the sensor datasheets and module pinouts.
- Ensure power supply and ground connections are properly set up.

4. Program the Arduino

1. Include Necessary Libraries:

- Open Arduino IDE.
- Go to Sketch > Include Library > Manage Libraries.
- Install the following libraries:
  - ESP8266WiFi or WiFi for ESP32
  - ThingSpeak
  - Any sensor-specific libraries (e.g., DHT for DHT sensors).

2. Write the Arduino Code:

- Below is a sample code to read data from a DHT11 sensor and send it to ThingSpeak:

```
cpp
#include <ESP8266WiFi.h>   // or #include <WiFi.h> for ESP32
#include <ThingSpeak.h>
#include <DHT.h>

// Replace with your network credentials
const char ssid = "your_SSID";
const char password = "your_PASSWORD";

WiFiClient client;
unsigned long myChannelNumber = YOUR_CHANNEL_NUMBER;
```

```
const char  myWriteAPIKey = "YOUR_WRITE_API_KEY";

#define DHTPIN D4          // Pin where the DHT sensor is connected
#define DHTTYPE DHT11      // DHT 11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  dht.begin();
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  ThingSpeak.begin(client);
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" C ");
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.println(" %");

  ThingSpeak.setField(1, t);
  ThingSpeak.setField(2, h);

  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  if (x == 200) {
    Serial.println("Channel update successful.");
  } else {
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  delay(20000); // ThingSpeak allows updates every 15 seconds
}
```

3. Upload the Code:

- Select the correct board and port from Tools > Board and Port.
- Click the upload button.

5. Monitor Data on ThingSpeak

1. Visualize Data:

- Go to your ThingSpeak channel.
- You should see real-time data updates and visualizations for your sensors.

2. Analyze Data:

- Use MATLAB Analysis and Visualizations in ThingSpeak for advanced data processing and insights.

6. Advanced Configuration

1. Create MATLAB Scripts:

- Go to Apps > MATLAB Analysis on ThingSpeak.
- Create scripts for advanced data analytics, machine learning models, and more.

2. Automate Responses:

- Set up ThingSpeak alerts or use the MATLAB code to trigger actuators based on sensor data (e.g., turning off lights when no motion is detected).

7. Final Setup and Deployment

1. Test Your System:

- Ensure all sensors and actuators work as expected.
- Fine-tune the system for optimal performance.

2. Deploy:

- Install the hardware in your home.
- Ensure a stable internet connection for continuous operation.

By following these steps, you'll have a functional Intelligent Home Automation System that uses AI and IoT to manage energy consumption and reduce your carbon footprint.