```
#include <DHT.h>
#include <Adafruit_NeoPixel.h>
#include <ArduinoBLE.h>

// 定义引脚
#define DHTPIN 11
#define DHTTYPE DHT22
#define LED_PIN 2
#define NUM_LEDS 1

// 初始化对象
DHT dht(DHTPIN, DHTTYPE);
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

// BLE服务和特征
BLEService ledService("19b10000-e8f2-537e-4f6c-d104768a1214");
BLECharacteristic controlCharacteristic("19b10001-e8f2-537e-4f6c-d104768a1214",
                                     BLEWrite | BLEWriteWithoutResponse, 20);

// 全局变量
int mode = 1;           // 1：温度控制模式，2：手动控制模式
int brightness = 255;
bool lightOn = true;
int currentColor = 1;    // 当前颜色（1-5）

// 温度范围和颜色名称
const float TEMP_RANGES[] = {25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5};
const char* COLOR_NAMES[] = {
   "Blue", "Cyan", "Green", "Yellow", "Orange", "Orange-Red", "Red"  // 温度模式颜色
};
const char* MANUAL_COLOR_NAMES[] = {
   "Red", "Green", "Blue", "Yellow", "Purple"  // 手动模式颜色
};

void setup() {
   Serial.begin(9600);
   Serial.println("Starting...");

   // 初始化BLE
   if (!BLE.begin()) {
       Serial.println("BLE启动失败!");
       while (1);
   }

   // 设置BLE
   BLE.setLocalName("Smart-Light");
   BLE.setAdvertisedService(ledService);
```

```
    ledService.addCharacteristic(controlCharacteristic);
    BLE.addService(ledService);
    BLE.advertise();
    Serial.println("BLE已准备就绪");

    // 初始化设备
    dht.begin();
    strip.begin();
    strip.show();

    // 初始化状态
    mode = 1;         // 默认温度模式
    brightness = 255;
    lightOn = true;
    currentColor = 1;
}

void loop() {
    BLEDevice central = BLE.central();

    if (central) {
        Serial.println("设备已连接: " + String(central.address()));

        while (central.connected()) {
            // 检查BLE命令
            if (controlCharacteristic.written()) {
                String command = String((char*)controlCharacteristic.value());
                handleCommand(command);
            }

            // 温度模式更新
            if (mode == 1 && lightOn) {
                updateTemperatureMode();
            }
        }

        Serial.println("设备已断开");
    }

    // 串口控制(调试用)
    if (Serial.available()) {
        String command = Serial.readStringUntil('\n');
        handleCommand(command);
    }

    delay(100);
}
```

```
void updateTemperatureMode() {
    static unsigned long lastUpdate = 0;
    if (millis() - lastUpdate >= 2000) {
        float temp = dht.readTemperature();
        float humi = dht.readHumidity();

        if (!isnan(temp) && !isnan(humi)) {
            Serial.println("\n----- 当前状态 -----");
            Serial.println("模式: 温度控制模式");
            Serial.println("温度: " + String(temp, 1) + "°C");
            Serial.println("湿度: " + String(humi, 1) + "%");

            // 设置对应的颜色
            setTemperatureColor(temp);

            printStatus();
        } else {
            Serial.println("温湿度读取失败");
        }
        lastUpdate = millis();
    }
}

void handleCommand(String command) {
    Serial.println("收到命令: " + command);

    // 模式切换命令
    if (command == "M1") {
        mode = 1;
        lightOn = true;
        Serial.println("切换到温度控制模式");
        return;
    }
    else if (command == "M2") {
        mode = 2;
        lightOn = true;
        Serial.println("切换到手动控制模式");
        setManualColor(String(currentColor));
        return;
    }

    // 手动模式下的命令
    if (mode == 2) {
        if (command.startsWith("L")) {
            // 亮度控制
            int newBrightness = command.substring(1).toInt();
            if (newBrightness == 0) {
                lightOn = false;
                strip.setBrightness(0);
                strip.show();
                Serial.println("灯光关闭");
            } else {
                lightOn = true;
                brightness = constrain(newBrightness, 0, 255);
                strip.setBrightness(brightness);
                setManualColor(String(currentColor));
                Serial.println("亮度设置为： " + String(brightness));
            }
        }
        else if (command.startsWith("C") && lightOn) {
            // 颜色控制
            currentColor = command.substring(1).toInt();
            setManualColor(String(currentColor));
            Serial.println("颜色设置为： " + String(currentColor));
        }
    }

    printStatus();
}

void setTemperatureColor(float temp) {
    if (!lightOn) return;

    uint32_t color;
    if (temp < TEMP_RANGES[0]) {
        color = strip.Color(0, 0, 255);  // 蓝色
    } else if (temp >= TEMP_RANGES[6]) {
        color = strip.Color(255, 0, 0);  // 红色
    } else {
        for (int i = 0; i < 6; i++) {
            if (temp < TEMP_RANGES[i + 1]) {
                switch(i) {
                    case 0: color = strip.Color(0, 255, 255); break;  // 青色
                    case 1: color = strip.Color(0, 255, 0); break;    // 绿色
                    case 2: color = strip.Color(255, 255, 0); break;  // 黄色
                    case 3: color = strip.Color(255, 165, 0); break;  // 橙色
                    case 4: color = strip.Color(255, 69, 0); break;   // 橙红色
                    default: color = strip.Color(255, 0, 0); break;   // 红色
                }
                break;
            }
        }
    }

    strip.setPixelColor(0, color);
    strip.setBrightness(brightness);
    strip.show();
}

void setManualColor(String colorNum) {
    if (!lightOn) return;

    uint32_t color;
    switch (colorNum.toInt()) {
        case 1:
            color = strip.Color(255, 0, 0);    // 红
            break;
        case 2:
            color = strip.Color(0, 255, 0);    // 绿
            break;
        case 3:
            color = strip.Color(0, 0, 255);    // 蓝
            break;
        case 4:
            color = strip.Color(255, 255, 0);  // 黄
            break;
        case 5:
            color = strip.Color(255, 0, 255);  // 紫
            break;
        default:
            color = strip.Color(255, 0, 0);    // 默认红色
            break;
    }

    strip.setPixelColor(0, color);
    strip.setBrightness(brightness);
    strip.show();
}

void printStatus() {
    Serial.println("\n----- 当前状态 -----");
    Serial.println("模式： " + String(mode == 1 ? "温度控制模式" : "手动控制模式"));
    Serial.print("灯光状态： ");
    Serial.println(lightOn ? "开启" : "关闭");
    Serial.print("亮度： ");
    Serial.println(brightness);
    if (mode == 2) {
        Serial.println("当前颜色： " + String(MANUAL_COLOR_NAMES[currentColor-1]));
    }
    Serial.println("-------------------");
}
```