Gesture based control of lighting system for different moods.

**Introduction:**

In today’s smart home environment, intuitive control systems enhance user experience. This project demonstrates a gesture-based lighting system that uses hand gestures to control various lighting moods: Relaxation, Nighttime, Romantic, and Focus. By using an APDS9960 gesture sensor, users can seamlessly switch between different lighting settings.

**Objectives:**

* To implement a gesture recognition system for controlling LED lights.
* To create distinct lighting moods based on specific gestures.
* To provide an intuitive and user-friendly interface for lighting control.

**Components Required:**

* Arduino Uno (or compatible board)
* APDS9960 Gesture Sensor
* White LEDs (for different moods)
* Resistors (220 ohm for LEDs)
* Breadboard and jumper wires
* Power supply (USB or battery)

**Hardware Setup:**

1. **Connect the APDS9960 Gesture Sensor**:
   * VCC to Arduino 5V
   * GND to Arduino GND
   * SDA to Arduino A4 (I2C data)
   * SCL to Arduino A5 (I2C clock)
2. **Connect the LEDs**:
   * Connect the LEDs to digital pins (12 and 14) with appropriate resistors in series.
3. **Power the Arduino**: Connect it to a power source.

Code Explanation Line by Line:

#include <Wire.h>

#include <Adafruit\_APDS9960.h>

// Gesture sensor object

Adafruit\_APDS9960 gestureSensor = Adafruit\_APDS9960();

// Pin definitions for White LEDs

const int ledRelaxationPin = 12; // Relaxation mood

const int ledNighttimePin = 14; // Nighttime mood

const int ledRomanticPin = 12; // Romantic mood

const int ledFocusPin = 14; // Focus/Work mood

1. **Include Libraries**: The Wire.h library enables I2C communication, while Adafruit\_APDS9960.h provides functions for the gesture sensor.
2. **Create Gesture Sensor Object**: An instance of the gesture sensor is created.
3. **Define LED Pins**: Constants are defined for the LED pins corresponding to different moods.

void setup() {

Serial.begin(9600);

// Initialize the APDS9960 sensor

if (!gestureSensor.begin()) {

Serial.println("Error initializing APDS9960 sensor!");

while (1);

}

Serial.println("APDS9960 sensor initialized.");

// Enable gesture detection

gestureSensor.enableGesture(true);

// Initialize LED pins as output

pinMode(ledRelaxationPin, OUTPUT);

pinMode(ledNighttimePin, OUTPUT);

pinMode(ledRomanticPin, OUTPUT);

pinMode(ledFocusPin, OUTPUT);

// Start with no LEDs on

turnOffAllLEDs();

}

4 **Setup Function**: Initializes serial communication for debugging and the gesture sensor. If initialization fails, it prints an error and halts.

5 **Enable Gesture Detection**: The sensor starts recognizing gestures.

6 **Pin Configuration**: Sets the LED pins as output.

7 **Turn Off All LEDs**: Ensures that all LEDs are off at the start.

void loop() {

// Check if a gesture is available

if (gestureSensor.gestureValid()) {

uint8\_t gesture = gestureSensor.readGesture();

switch (gesture) {

case APDS9960\_UP:

Serial.println("Gesture: Up");

turnOffAllLEDs(); // Turn off all LEDs

digitalWrite(ledFocusPin, HIGH); // Turn on Focus LED

break;

case APDS9960\_DOWN:

Serial.println("Gesture: Down");

turnOffAllLEDs(); // Turn off all LEDs

digitalWrite(ledNighttimePin, HIGH); // Turn on Nighttime LED

break;

case APDS9960\_LEFT:

Serial.println("Gesture: Left");

turnOffAllLEDs(); // Turn off all LEDs

digitalWrite(ledRomanticPin, HIGH); // Turn on Romantic LED

break;

case APDS9960\_RIGHT:

Serial.println("Gesture: Right");

turnOffAllLEDs(); // Turn off all LEDs

digitalWrite(ledRelaxationPin, HIGH); // Turn on Relaxation LED

break;

default:

Serial.println("No gesture detected");

break;

}

}

}

**9 Loop Function**: Continuously checks for gestures. When a valid gesture is detected, it identifies the gesture using a switch-case statement and activates the corresponding LED while turning off others.

// Function to turn off all LEDs

void turnOffAllLEDs() {

digitalWrite(ledRelaxationPin, LOW);

digitalWrite(ledNighttimePin, LOW);

digitalWrite(ledRomanticPin, LOW);

digitalWrite(ledFocusPin, LOW);

}

**10 Turn Off All LEDs**: A utility function that turns off all LEDs, ensuring that only one mood is active at a time.

**Future Improvements:**

* Implement more mood settings with additional LEDs.
* Enhance gesture recognition for more nuanced controls (e.g., double tap, hold).
* Integrate with a mobile app for additional controls and customizations.
* Use RGB LEDs for more diverse lighting options and effects.

**Conclusion:**

This project illustrates a simple yet effective way to control a lighting system using gesture recognition. By utilizing the APDS9960 sensor and Arduino, users can effortlessly switch between different lighting moods, enhancing their living spaces with a touch of modern technology. The project can be further expanded with additional features and improvements for a more comprehensive smart home experience.

ANUSUYA K

ECE