NATIONAL INSTITUTE OF TECHNOLOGY AGARTALA

ELECTRICAL & ELECTRONIC INSTRUMENTATION AND MEASUREMENT LAB

ELECTRONICS AND INSTRUMENTATION ENGINEERING

SUBMITTED BY:

GROUP 2:

ANKIT KUMAR SINGH 21UEI006

MUSKAN GUPTA 21UEI007

GOLU KUMAR 21UEI008

ASHISH KUMAR 21UEI009

ABANTIKA DEB 21UEI010

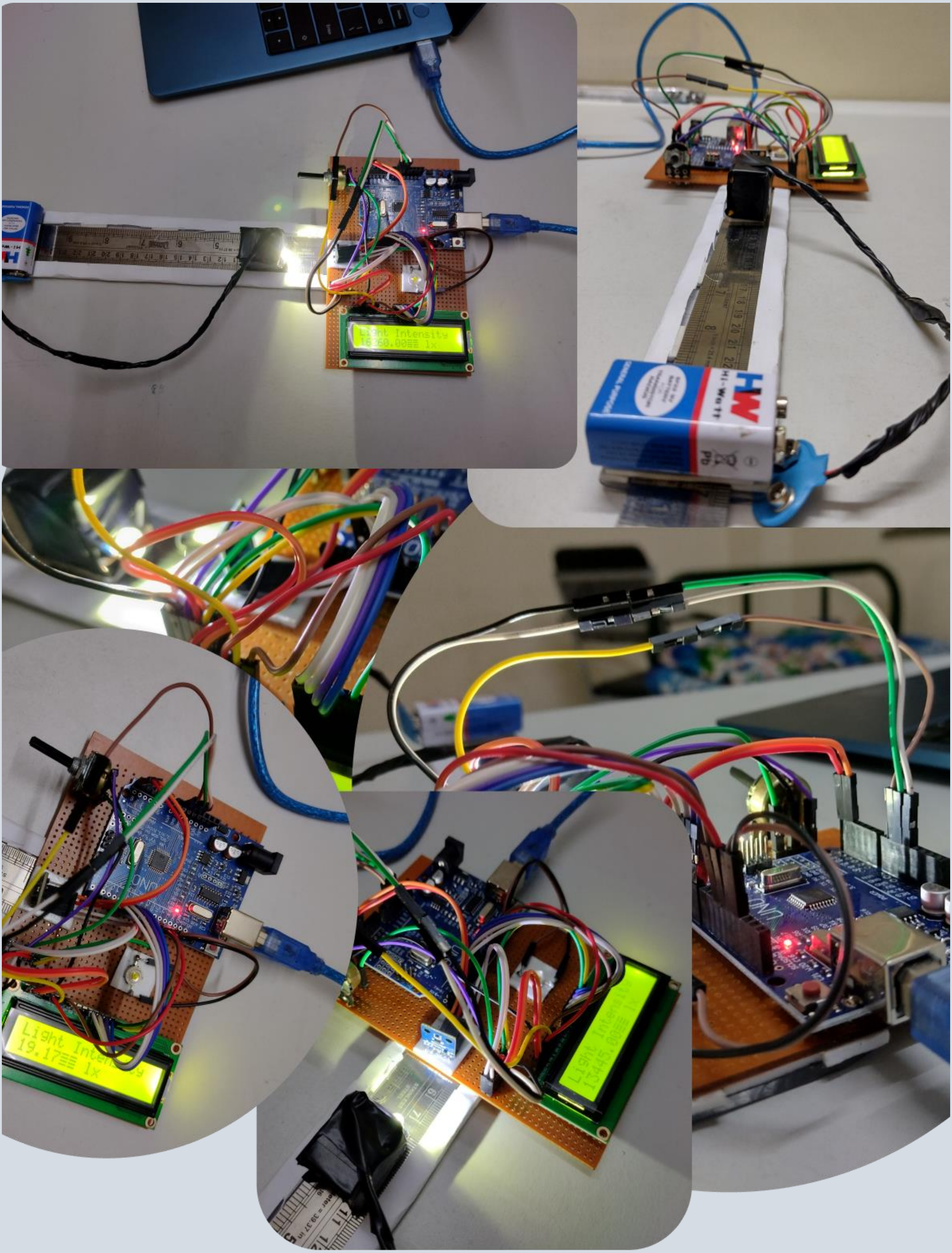
IN THE GUIDENCE OF:

DR. SUBIR DAS (ASSISTANT PROFESSOR)

DR. DEBANJAN ACHARYYA (ASSISTANT PROFESSOR)

MR. DILIP BISWASARMA (TECHNICAL ASSISTANT)

MR. KAILASH PRATIM PHUKAN (TECHNICAL ASSISTANT)



LIGHT INTENSITY DETECTION SYSTEM

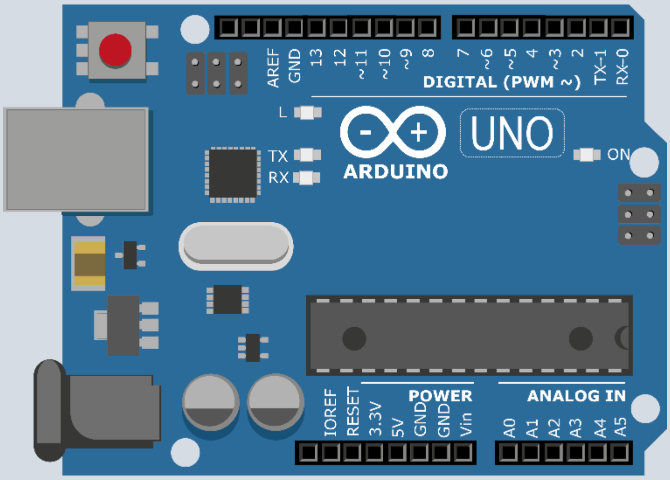
Introduction

The Smart Light Control System is a project designed to enhance energy efficiency and user convenience by automatically adjusting lighting intensity based on ambient light levels. Equipped with light sensors and a central microcontroller, this system continually monitors the surrounding light conditions and makes real-time decisions. When ambient light is very low, the system turns on the lights to an appropriate brightness level for optimal visibility, and it adapts to changing conditions, dimming or brightening the lights to maintain a comfortable environment. This results in significant energy savings, as lights are only on when necessary. Users also have manual control options to adjust lighting as per their preferences. The project has applications in homes, offices, street lighting, and industrial facilities, offering an intelligent, energy-efficient, and user-friendly solution for lighting control, contributing to energy conservation and enhanced lighting experiences.

Components

➢ Arduino Uno R3:

The Arduino Uno R3 plays a central role in the project described, which detects surrounding light intensity and controls lighting based on the ambient conditions. Here are the key roles of the Arduino Uno R3 in this project:



- **Microcontroller:** The Arduino Uno R3 serves as the microcontroller for the project. It processes data from the light sensor and controls the LED based on the surrounding light intensity. It acts as the "brain" of the system, making real-time decisions about whether to turn the LED on or off.
- **Data Processing:** The Arduino Uno R3 reads the data from the BH1750 light sensor and converts it into a usable format. It then compares the light intensity value with a predefined threshold (15 lux in this case) to determine whether the LED should be turned on or off.
- **Control Logic:** The Arduino Uno R3 contains the logic for the conditional statements that determine the LED's state. It checks if the ambient light intensity is low (below or equal to 15 lux) and, based on this condition, sends a signal to either turn the LED on (HIGH) or turn it off (LOW).
- **Output Control:** The microcontroller is responsible for controlling the digital output pin (Pin 13 in this case) connected to the LED. It sends appropriate signals to this pin to control the LED's state, ensuring that it is in sync with the surrounding light conditions.
- **Serial Communication:** The Arduino Uno R3 is also used for serial communication. It sends the current light intensity values to a connected computer or device via a serial connection, allowing real-time monitoring and data logging if required.
- **User Interface:** While not explicitly mentioned in the provided code, the Arduino Uno R3 can be used to interface with additional input devices or user interfaces, allowing users to adjust the threshold or override the automatic lighting control as needed. For instance, you could incorporate physical buttons or a smartphone app to interact with the system.

➢ BH1750 Light Sensor:

The BH1750 light sensor plays a crucial role in the project that detects surrounding light intensity and controls lighting based on ambient conditions. Here are the key roles of the BH1750 sensor in this project:

- **Light Sensing:** The primary role of the BH1750 sensor is to measure the intensity of the surrounding light. It uses a photodetector to detect the amount of light falling on its surface and provides a digital lux value, which is a measure of the illuminance or light intensity.
- **Data Acquisition:** The BH1750 sensor continuously collects data about the ambient light conditions. It converts the analog light intensity into a digital format, making it suitable for processing by the Arduino Uno R3.
- **Real-Time Updates:** The sensor provides real-time updates about changes in light intensity, allowing the Arduino to make immediate decisions based on the current lighting conditions.
- **Threshold Comparison:** The Arduino uses the lux value obtained from the BH1750 sensor to compare it with a predefined threshold (e.g., 15 lux). This comparison determines whether the surrounding light intensity is low (below the threshold) or high (above the threshold).
- **Automation:** Based on the lux value from the BH1750 sensor, the Arduino can automatically control the lighting system by turning the LED on when the surrounding light is very low (below the threshold) and turning it off when the light intensity is high or normal (above the threshold).
- **Data Logging**: While not explicitly shown in the provided code, the data from the BH1750 sensor can be used for data logging and analysis, allowing users to monitor light intensity trends over time.

Pin Configuration:

The BH1750 light sensor typically comes in a small module with three pins. Here is the pin configuration for the BH1750 sensor module:

- **VCC (Power Supply):** This is the power supply pin. It should be connected to a 3.3V or 5V power source, depending on the sensor's voltage requirement.
- **SDA (Serial Data**): This is the serial data pin. It is used for I2C communication and should be connected to the SDA (data) pin on your microcontroller, such as an Arduino.
- **SCL (Serial Clock):** This is the serial clock pin. It is used for I2C communication and should be connected to the SCL (clock) pin on your microcontroller.
- **ADDR:** Device address pin, used to select the address when more than two modules are connected.

➢ LCD Display 16*2:

A typical 16x2 LCD display module, commonly used in various electronic projects, is often based on the HD44780 controller, which is widely supported in the Arduino community. The pin diagram of a 16x2 LCD display typically includes 16 pins, which are divided into two rows of 8 pins each. Here's an explanation of the pinout:

First Row (Top Row) - Pins 1 to 8 (from left to right):

- **VSS (Ground):** Connect this pin to the ground (0V) of your circuit.
- **VCC (Power):** Connect this pin to the supply voltage, typically +5V or +3.3V.
- **VEE (Contrast Control):** This pin controls the contrast of the display. You can connect it to a potentiometer (a variable resistor) to adjust the contrast.
- **RS (Register Select):** This pin determines whether you're sending data or a command to the display. Connect it to the Arduino to specify whether you're sending character data (RS=1) or an instruction command (RS=0).
- **R/W (Read/Write):** This pin allows you to read from the LCD (R/W=1) or write to it (R/W=0). In most cases, you set it to write mode (R/W=0) because you're primarily sending data to the display.
- **E (Enable):** The Enable pin is used to enable the LCD for data or command execution. A high-to-low transition (or low-to-high, depending on configuration) on this pin triggers the data/command transfer.
- **D0 to D7 (Data Pins):** These are the data pins. These pins carry the 8-bit data to be displayed on the LCD. Typically, you can use 4-bit mode to reduce the number of pins used, connecting D4 to D7, but in the 8-bit mode, all 8 data pins are used for higher data transfer speed.

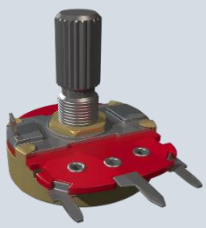
Second Row (Bottom Row) - Pins 9 to 16 (from left to right):

- **LED+ (Backlight Power):** Connect this pin to the positive terminal of your backlight LED. It's usually connected to +5V.
- **LED- (Backlight Ground):** Connect this pin to the ground terminal of your backlight LED.

Note: Pins 10 to 15 are not commonly used in standard 16x2 LCD displays and are often left unconnected.

➢ Potentiometer:

In this project, a potentiometer is employed to control the contrast of the 16x2 LCD display. The potentiometer allows users to fine-tune the contrast setting, which is crucial for ensuring the readability of characters and values on the display under varying lighting conditions. By adjusting the potentiometer, users can optimize the contrast to make the text visible and crisp, irrespective of whether the environment is well-lit or dimly lit. This user-adjustable feature enhances the usability and adaptability of the project, ensuring that information on the LCD remains legible and clear in a range of ambient lighting scenarios.



➢ LED:

The LED in this project serves as a practical indicator of the system's operation. It turns on when the ambient light intensity falls below a specified threshold (e.g., 15 lux), and it turns off when the light intensity is high or normal. This LED provides a visual cue to users, signaling whether the system has determined that artificial lighting is needed in response to low ambient light conditions or not. It enhances user awareness and demonstrates the real-time control of lighting based on environmental factors, contributing to energy savings and convenience.

➢ 9V Battery:

A 9V battery is used in the project to power the light source. The light source mounted on the slider connected to a battery.

➢ Slider:

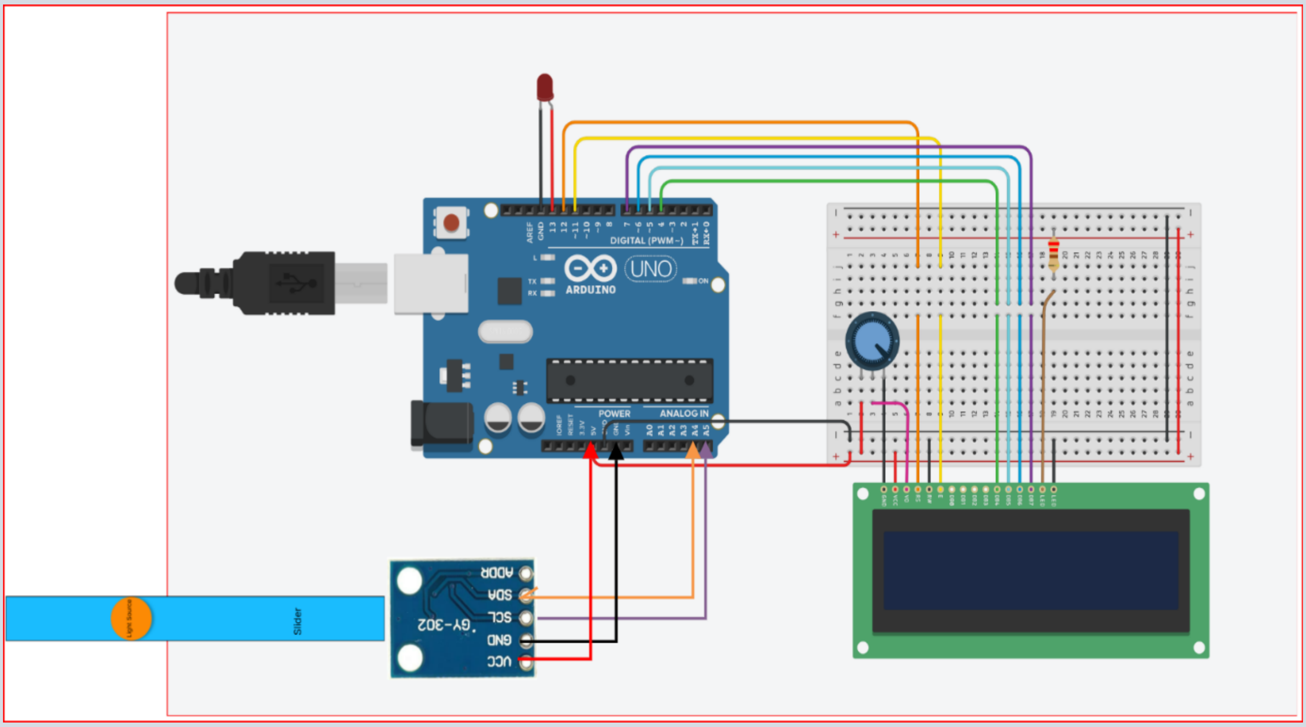
A slider is placed in front of BH1750 light sensor. It has a movable light source mounted on it. When the light source moves forward the intensity of light increases and displayed in LCD display. When it is moved backward the intensity of light decreases and displayed in LCD display.

Arduino Code

```
#include <Wire.h>       // adds I2C library
#include <BH1750.h>      // adds BH1750 library file
#include <LiquidCrystal.h> // includes the LiquidCrystal Library
LiquidCrystal lcd(12, 11, 4, 5, 6, 7); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)
BH1750 lightMeter;
#define led 13

void setup() {
  Serial.begin(9600);       // set serial baud rate of 9600
  Wire.begin();           // Enable I2C pins of Arduino
  lcd.begin(16, 2);         // Defines LCD size to 16x2
  lcd.print("Light Intensity");   //Print "Light Intensity" message on first line
  lightMeter.begin();
  pinMode(led,OUTPUT);
  delay(2000);
}
void loop() {
 float lux = lightMeter.readLightLevel();
 if(lux<=15){
  digitalWrite(led,HIGH);
 }
 else{
  digitalWrite(led,LOW);
 }
  lcd.setCursor(0, 1);
  lcd.println(lux);
  Serial.println(lux);
  lcd.print(" lx");
  delay(1000);
}
```

Circuit Diagram



Benefits:

1. **Energy Efficiency**: The project optimizes energy consumption by automatically turning on lights when ambient light is low and turning them off when it's sufficient, reducing electricity usage and lowering energy costs.

2. **User Convenience**: Users do not need to manually control the lights as the system adapts to changing light conditions, providing a hassle-free and convenient lighting experience.

3. **Enhanced Visibility**: The system ensures that rooms are adequately illuminated during low-light conditions, improving visibility and comfort for occupants.

4. **Adaptability**: Users can customize the threshold for light intensity based on their preferences and specific requirements.

5. **Real-time Monitoring**: The LCD display provides real-time feedback on light intensity, aiding users in understanding the system's operation.

6. **DIY Home Automation**: Ideal for DIY enthusiasts, this project offers an affordable and accessible way to create a home automation solution for smart lighting control.

Conclusion:

In conclusion, the project that detects surrounding light intensity and controls lighting based on ambient conditions offers a cost-effective and energy-efficient solution for various applications. It seamlessly adapts to changing light levels, enhancing user convenience and reducing energy consumption. The real-time monitoring through the LCD display and the adaptability of the system's threshold provides users with control and insight. Overall, this project not only contributes to significant energy savings and cost efficiency but also aligns with the growing emphasis on sustainable, smart technologies. Its DIY-friendly nature makes it accessible to a broad range of users seeking to improve lighting efficiency and user experiences in diverse settings.