AUTO INTENSITY CONTROL OF STREET LIGHTS

A

PROJECT REPORT

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATIONS ENGINEERING

SUBMITTED BY:

AYUSH, VAIBHAV, VISHAL

UE215107, UE215106, UE215109

SIXTH SEMESTER

UNDER THE SUPERVISION OF:

AMIT CHAUDHARY

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY,

PANJAB UNIVERSITY, CHANDIGARH, INDIA

May 2024

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY,

PANJAB UNIVERSITY, CHANDIGARH

CANDIDATE'S DECLARATION

We hereby certify that the work presented in the Project entitled "**Auto Intensity Controls of Street Lights**" in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in Electronics and Communications Engineering from University Institute of Engineering and Technology, Panjab University, Chandigarh, is an authentic record of our own work carried out under the supervision and guidance of (Name(s), Designation of supervisors).

Date: 09-09-2024
Place: Chandigarh

Ayush, Vaibhav, Vishal
UE215107, 106, 109
(Six Semester)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 09-09-2024
Place: Chandigarh

Amit Chaudhary
Associate Professor, ECE
UIET, Panjab University,
Chandigarh

ABSTRACT

The "Auto Intensity Control Street Light" system is an innovative, Arduino-based solution designed to optimize street lighting by automatically adjusting brightness in response to real-time environmental conditions. By integrating components such as Light Dependent Resistors (LDRs), LEDs, an LCD, and a pushbutton, the system ensures energy-efficient and safe street lighting, adapting to varying light levels throughout the day.

The system's core functionality is driven by the Arduino microcontroller, which continuously monitors ambient light via LDR sensors. This data allows the system to automatically adjust the intensity of LED street lights, ensuring optimal illumination. The LCD provides real-time feedback, allowing users to monitor system performance, while a pushbutton offers manual control for flexibility, enabling adjustments based on specific needs.

Overall, this system offers a significant advancement in urban lighting by reducing energy consumption, lowering maintenance costs, and minimizing light pollution. It aligns with smart city initiatives, providing a scalable, cost-effective, and environmentally sustainable solution for modernizing urban infrastructure.

TABLE OF CONTENTS

1. INTRODUCTION

The "Auto Intensity Control Street Light" system offers a smart solution to these challenges by using modern technology to dynamically adjust the brightness of street lights based on real-time environmental data. The system integrates an Arduino microcontroller, Light Dependent Resistors (LDRs), energy-efficient Light Emitting Diodes (LEDs), a Liquid Crystal Display (LCD), and a pushbutton for manual control. By continuously monitoring ambient light levels with LDRs, the Arduino adjusts the LED brightness to ensure optimal illumination, conserving energy when full brightness is unnecessary. The system's LCD provides real-time status updates, while the pushbutton allows users to manually override automated settings, offering both efficiency and flexibility in urban street lighting management.

1.1 OBJECTIVE

The primary objective of the Auto Intensity Control of Street Lights project using Arduino Uno is to develop an intelligent and energy-efficient street lighting system that automatically adjusts the brightness of street lights based on real-time environmental conditions. Specifically, the project aims to achieve the following:

1. **Energy Optimization**: To reduce unnecessary power consumption by dynamically adjusting the intensity of street lights in response to ambient light levels, ensuring that lights operate at full brightness only when needed.

2. **Enhanced Safety and Visibility**: To maintain adequate illumination during low-light conditions or when motion is detected, thereby improving safety and visibility for pedestrians and vehicles.

3. **User Control and Interaction**: To provide users with the ability to manually override and adjust light intensity settings through a pushbutton interface, ensuring flexibility and control over the lighting system.

4. **Sustainability**: To contribute to environmental conservation by minimizing energy usage, lowering maintenance costs, and reducing light pollution, aligning with the goals of sustainable urban development.

1.2 FUTURE SCOPE:

1. **Smart City Integration**: Combining with IoT and AI for real-time adjustments and centralized control.

2. **Sustainability**: Utilizing renewable energy sources like solar power and energy harvesting technologies.

3. **Advanced Sensing**: Adaptive lighting based on conditions and data analytics for optimizing energy use.

4. **Safety Enhancements**: Integration with emergency response systems and surveillance support

2. Components and Specifications:

2.1 Arduino Uno

Description:
The Arduino Uno is a microcontroller board based on the ATmega328P. It's a popular choice for beginners and hobbyists due to its simplicity and versatility.

Specifications:

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- Flash Memory: 32 KB (of which 0.5 KB is used by the bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- USB Connection: USB-B
- Dimensions: 68.6 mm x 53.4 mm
- Weight: Approximately 25 g

2.2 LDR Sensor (Light Dependent Resistor)

Description:
An LDR, or Light Dependent Resistor, is a light-sensitive resistor whose resistance decreases with increasing incident light intensity. It's used for light sensing applications.

Specifications:

- Resistance (at low light): High, typically 1 MΩ or more
- Resistance (at high light): Low, typically 1 kΩ or less
- Wavelength Range: Typically, 400-700 nm (visible light)
- Response Time: Varies, generally in the range of milliseconds
- Package Type: Often cylindrical or rectangular with two leads

2.3 LCD (Liquid Crystal Display) - Common Example: 16x2 LCD

Description:
A 16x2 LCD is a display module that can show 16 characters per line and has 2 lines. It uses liquid crystal technology to display text.

Specifications:

- Display Size: 16 characters x 2 lines
- Controller IC: HD44780 or compatible
- Operating Voltage: 5V (common) or 3.3V (some versions)
- Backlight: Typically LED backlit (can be on or off)
- Interface: 4-bit or 8-bit parallel
- Dimensions: Varies depending on the manufacturer
- Character Size: Typically, 5x8 pixels

2.4 LED (Light Emitting Diode)

Description:
An LED is a semiconductor light source that emits light when current flows through it. They are used for indication and display purposes.

Specifications:

- Forward Voltage: Typically, 2V to 3.3V (depending on color)
- Forward Current: Typically, 10 mA to 20 mA
- Wavelength: Varies with color (e.g., Red: 620-630 nm, Green: 520-530 nm)
- Luminous Intensity: Varies based on type and manufacturer
- Package Type: Usually through-hole or surface-mount
- Viewing Angle: Typically, 20° to 60°

2.5. Pushbutton

Description:
A pushbutton is a mechanical switch that opens or closes an electrical circuit when pressed. It’s commonly used for user inputs.

Specifications:

- Contact Rating: Varies, often up to 50V DC, 0.5A
- Actuation Force: Varies by design, usually between 1N to 5N
- Switch Type: Momentary (returns to default position when released)
- Mounting Type: Through-hole or surface-mount
- Dimensions: Varies with design
- Lifespan: Typically rated for millions of cycles

3 Working Principle and Project Image:

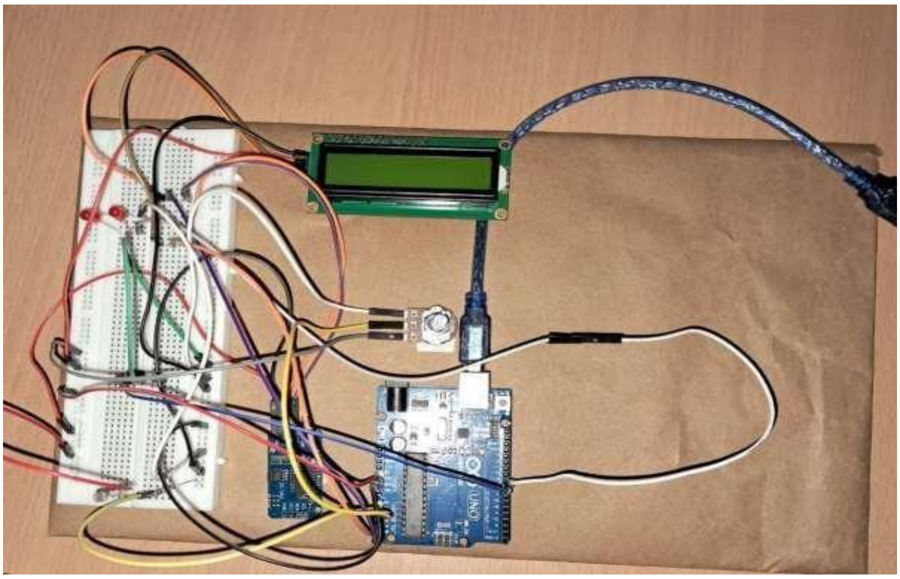

Fig: Auto Intensity Control of Street Light

(Project Image)

3.1 Working Principle:

3.1.1 Ambient Light Measurement:

- LDR Setup: The LDR is connected to an analog input pin on the Arduino. It’s part of a voltage divider circuit with a fixed resistor.
- Voltage Divider: The LDR's resistance decreases with increasing light intensity, causing the voltage across the LDR to change. This voltage is read by the Arduino as an analog value.

3.1.2 Data Processing:

- Analog Reading: The Arduino reads the voltage from the LDR sensor and converts it into a corresponding digital value (light intensity level).
- Threshold Calculation: The Arduino compares the light intensity value with predefined threshold levels to determine the appropriate brightness for the street light.

3.1.3 Control Signal Generation:

- PWM Control (if using dimmable LEDs): The Arduino generates a Pulse Width Modulated (PWM) signal to adjust the brightness of the LED. Higher PWM duty cycles increase brightness, while lower duty cycles decrease it.
- Relay Control (for on/off control): If using a standard LED or light bulb that cannot be dimmed via PWM, the Arduino can control a relay to turn the street light on or off based on the light intensity.

3.1.4 Automatic Adjustment:

- Day/Night Control: During the day, the ambient light is high, so the street light is turned off or set to low intensity. At night, when ambient light is low, the street light is turned on or set to higher intensity.
- Intensity Adjustment: The system automatically adjusts the light intensity of the street light based on the LDR readings, ensuring the light level is appropriate for the current ambient conditions.

3.1.5 Manual Override (Optional):

- Pushbutton: A pushbutton can be connected to the Arduino to allow manual control or override of the automatic settings. For example, pressing the button to toggle the street light between automatic and manual modes or adjustintensity levels manually.

4. Implementation and Testing

Implementing and testing an automated intensity control system for street lights involves several steps, from hardware assembly to programming and final testing:

4.1. Components Use:

- Arduino Uno
- LDR Sensor
- LED or Light Bulb
- Relay Module (if controlling high-power light bulbs)
- Resistors (for LDR voltage divider)
- Pushbutton (optional for manual control)
- Breadboard and Jumper Wires
- Power Supply

4.2. Build the Circuit:

a. LDR Circuit:

- Connect one terminal of the LDR to the 5V pin on the Arduino.
- Connect the other terminal of the LDR to an analog input pin (e.g., A0) on the Arduino.
- Connect a fixed resistor (e.g., 10kΩ) from the analog input pin to ground (GND).

b. LED Control:

- If using an LED, connect the anode (longer lead) to a PWM-capable digital pin (e.g., D9) through a current-limiting resistor (e.g., 220Ω).
- Connect the cathode (shorter lead) to ground (GND).

c. Relay Module:

- Connect the relay control pin to a digital pin (e.g., D8) on the Arduino.
- Connect the common terminal (COM) of the relay to one terminal of the light bulb or high-power light source.
- Connect the normally open terminal (NO) of the relay to the power supply of the light bulb.
- Connect the other terminal of the light bulb to the ground of the power supply.

d. Pushbutton (Optional):

- Connect one terminal of the pushbutton to a digital input pin (e.g., D2) on the Arduino.
- Connect the other terminal to ground (GND).
- Use the internal pull-up resistor in the Arduino (by setting the pin mode to INPUT_PULLUP).

4.3. Write and upload the Arduino Code:

. Connect the Arduino to your computer using a USB cable.

. Open the Arduino IDE and paste the code into a new sketch.

. Select the appropriate board and port from the Tools menu.

. Upload the code to the Arduino by clicking the upload button.

4.4. Testing:

a. Verify Connections:

- Ensure all components are connected correctly according to the circuit diagram.

b. Initial Power-Up:

- Power up the Arduino and observe the behavior of the LED and relay. Ensure that the LED is connected properly and the relay is functioning as expected.

c. Check LDR Readings:

- Open the Serial Monitor in the Arduino IDE to observe the LDR readings. You should see the light intensity values being printed.

d. Test Automatic Control:

- In Low Light: Cover the LDR to simulate a low-light condition. The LED should turn on or increase in brightness if you're using PWM control. The relay should also close to power the light bulb.
- In Bright Light: Expose the LDR to bright light. The LED should turn off or dim, and the relay should open to turn off the light bulb.

e. Test Manual Control (if using a pushbutton):

- Press the pushbutton to toggle between manual and automatic modes. Verify that in manual mode, you can override automatic settings and manually control the light.

f. Adjust Threshold:

- Modify the threshold value in the code if necessary to better suit your specific ambient light conditions.

  Troubleshooting Tips:

- No Light from LED or Bulb:
  - Check all connections and ensure the relay is wired correctly.
  - Ensure the power supply is adequate for the light bulb.
- Incorrect LDR Readings:
  - Verify the resistor values in the LDR voltage divider.
  - Check that the LDR is exposed to varying light conditions correctly.
- Pushbutton Not Working:
  - Confirm the pushbutton connections and ensure the pin mode is set to INPUT_PULLUP

5. Conclusion

The automated intensity control of street lights project is a practical and efficient application of microcontroller technology for managing lighting systems based on ambient light conditions. The automated intensity control system for street lights is a versatile and effective solution that showcases the power of integrating sensors and microcontrollers. By automating the control of street lighting based on ambient light levels, this system promotes energy efficiency, enhances safety, and offers practical applications for smart city development. Whether for a small-scale project or as a prototype for larger implementations, this project provides a strong foundation for future advancements in intelligent lighting systems.

We would like to express our deepest gratitude to Sharmili Mam for her continuous support and guidance throughout this project. We also extend our thanks to the Electronics and Communication Department of UIET, Panjab University, for providing us with the necessary resources and environment to complete this project.

5.1 References

1. Arduino. "Arduino Uno Rev3." Retrieved from

https://store.arduino.cc/products/arduino-uno-rev3

2.Arduino Reference. "Liquid Crystal Library." Retrieved from https://www.arduino.cc/en/Reference/LiquidCrystal

3. Auto Intensity Control of Street Lights using Arduino

https://www.electronicshub.org/auto-intensity-control-of-street-lights-using-arduino/