International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

# SMART GARDEN LIGHT USING ARDUINO UNO AND LDR SENSOR

*Prof. R.P.Khanapure ,Kamble Antima , Hodade Megharani ,Rashmi Suryawanshi*
*Professor, M S Bidve Engineering College, Latur, Maharashtra*
*234 Student at Electronics Engineering Department,*
*M.S. Bidve Engineering College, Latur, Maharashtra, India*
*D.R. Babasaheb Ambedkar Technological University, Lonere.*
*Email: antimakamble@gmail.com*

**Abstract**— The Smart Garden Light is an automatic lighting system designed to control garden lights according to the surrounding light intensity. The system uses an Arduino Uno microcontroller and a Light Dependent Resistor (LDR) sensor to detect the amount of natural light present in the environment. When sufficient daylight is available, the garden LEDs remain OFF. When the surrounding light becomes low, the Arduino automatically increases the brightness of the LEDs. This system helps reduce unnecessary electricity consumption and provides automatic garden illumination without requiring manual operation.

The project uses an LDR sensor, Arduino Uno, RGB LEDs, resistors, and a regulated power supply. The system is simple, cost-effective, and suitable for small gardens and outdoor lighting applications.

**Keywords:** Arduino Uno, Smart Garden Light, LDR Sensor, Automatic Lighting, RGB LED, Energy Saving.

## I. INTRODUCTION

Lighting is an important requirement in gardens and outdoor areas. Traditional garden lights are usually operated manually or remain switched ON for a fixed period. This can lead to unnecessary electricity consumption.To overcome this problem, the Smart Garden Light system is designed using an Arduino Uno and an LDR sensor. The LDR detects the surrounding light intensity and sends the corresponding signal to the Arduino. The Arduino processes the sensor value and controls the brightness of the LEDs. During bright daylight, the LEDs remain OFF. As the environment becomes darker, the LED brightness increases automatically. This project provides an easy and economical solution for automatic garden lighting.

## II. RELATED WORK

Automatic lighting systems are used in homes, streets, gardens, and other outdoor areas. Many conventional systems use manual switches or timers to control lighting. Light-dependent resistor-based systems provide another method of controlling lights according to environmental brightness. By connecting an LDR sensor to a microcontroller, the lighting system can respond automatically to changes in natural light. The proposed Smart Garden Light uses an Arduino Uno to read the LDR sensor and control RGB LEDs according to the detected light intensity.

## III. OBJECTIVES OF THE PROJECT

The main objectives of this project are:

1. To design an automatic garden lighting system.
2. To detect surrounding light intensity using an LDR sensor.
3. To control LED brightness using an Arduino Uno.
4. To reduce unnecessary electricity consumption.
5. To eliminate the need for manual switching.
6. To develop a simple and cost-effective lighting solution.

## IV. PROPOSED SYSTEM

The proposed system consists of an Arduino Uno, an LDR sensor, RGB LEDs, resistors, and a regulated power supply. The LDR sensor continuously detects the surrounding light intensity. Its output is connected to the analog input pin A0 of the Arduino. The Arduino reads the sensor value and compares it with predefined brightness levels. According to the detected light intensity, the Arduino controls the RGB LEDs connected to PWM pins D9, D10, and D11.
When the environment is bright, the LEDs remain OFF. When the environment becomes dark, the LED brightness increases automatically.

## IV. SYSTEM ARCHITECTURE AND CIRCUITRY

### 4.1 System Architecture

The proposed architecture consists of five principal stages:

1. Power supply section.
2. Ambient-light sensing section.
3. Microcontroller processing section.
4. PWM brightness-control section.
5. LED output section.

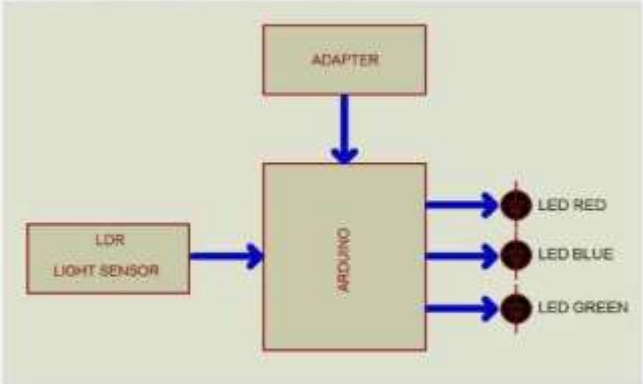
The LDR sensor detects changes in surrounding light intensity and produces a corresponding voltage signal. The Arduino Uno reads the signal through analog input A0 and processes it using the uploaded program. The



© 2026, IJSREM  DOI: 10.55041/IJSREM69238 

controller calculates the required brightness and sends PWM signals to the red, green, and blue LED channels through pins D9, D10, and D11.

### 4.2 FIG.BLOCKDIAGRAM




### 4.3 Power Supply and Voltage Regulation

A regulated 12 V DC adapter is used as the main power source. It connects to the DC barrel jack of the Arduino Uno. The Arduino provides a regulated 5 V supply for the LDR voltage divider and suitable low-power demonstration circuit.The external 12 V supply must not be connected directly to the Arduino 5 V pin, analog input, or LED-control outputs. A common ground reference is required for the Arduino and the connected LED circuits.

### 4.4 LDR Sensor Circuit

The sensing section consists of an LDR and a 10 kΩ resistor connected as a voltage divider. In the proposed circuit, the LDR connects toward the positive supply, while the 10 kΩ resistor connects toward ground. Their junction is connected to analog input A0.This arrangement converts changes in the LDR resistance into a measurable voltage. When more light falls on the LDR, its resistance decreases, raising the voltage at A0. When the surrounding light decreases, its resistance increases, lowering the voltage at A0.The Arduino converts the input voltage into a digital reading ranging from 0 to 1023.

### 4.5 Arduino Uno Control Unit

The Arduino Uno acts as the main processing unit of the project. It repeatedly reads the analog sensor value from pin A0 and determines the surrounding lighting condition. The program compares the sensor reading with predefined calibration values. It then calculates the required LED brightness and sends the corresponding PWM signals to the output pins. The Arduino Uno provides the analog inputs and PWM-capable digital outputs required for this project.

### 4.6 PWM Brightness Control

PWM stands for Pulse Width Modulation. It is a technique used to control the average power delivered to an electronic device by switching a digital output ON and OFF rapidly. A small duty cycle produces lower average LED power and dim illumination, while a larger duty cycle produces brighter illumination. The Arduino uses the analogWrite() function to generate PWM signals.

The output value ranges from 0 to 255:

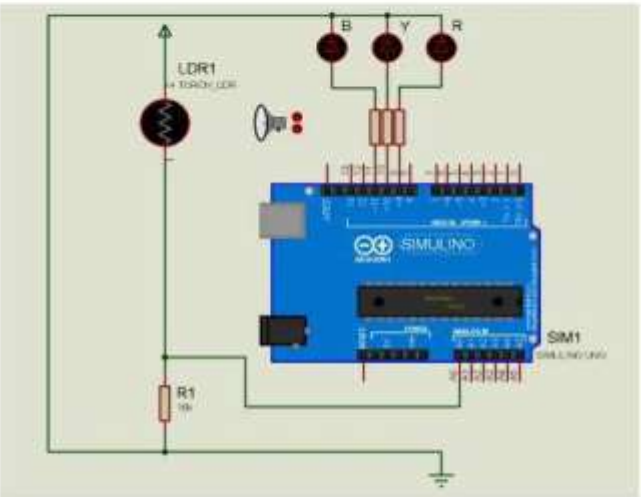
- 0: LED OFF.
- 80: Low brightness.
- 150: Medium brightness.
- 200: High brightness.
- 255: Maximum brightness.

The same PWM command can be applied to all three LED channels so that their brightness changes together.

### 4.7 LED Output Circuit

The output section consists of three LED channels: red, green, and blue. The red LED connects to PWM pin D9, the green LED connects to D10, and the blue LED connects to D11.Each LED uses an individual current-limiting resistor. The LEDs operate according to the PWM signals generated by the Arduino. These three channels represent the garden-lighting output of the prototype.

Fig. 2. Circuit Diagram of Smart Garden Light Using Arduino Uno and LDR Sensor




## V. CIRCUIT CONNECTIONS AND COMPONENT CONFIGURATION

### 5.1 Circuit Connections

The main electrical connections are given below.

| Component | Arduino Pin |
|---|---|
| LDR voltage-divider junction | A0 |
| Red LED | D9 |
| Green LED | D10 |

| | |
|---|---|
| Blue LED | D11 |
| LDR voltage divider | 5V and GND |
| LED circuit return | GND |
| DC adapter | DC barrel jack |

### 5.2 Power Management Matrix

**Table I: Power Management Summary**

| Component | Supply | Function |
|---|---|---|
| DC adapter | 12 V DC | Main power source |
| Arduino Uno | Regulated board supply | Controller operation |
| LDR circuit | 5 V | Light sensing |
| Red LED | Arduino PWM output | Lighting channel |
| Green LED | Arduino PWM output | Lighting channel |
| Blue LED | Arduino PWM output | Lighting channel |
| 10kΩ resistor | Voltage-divider circuit | Sensor signal generation |

The current consumption depends on the selected components and LED ratings. The supply capacity and resistor values should be selected according to their datasheets.

### 6.1 Algorithmic Workflow

After initialization, the Arduino repeatedly reads the LDR sensor connected to analog pin A0.The program takes 16 sensor readings and calculates their average to reduce small fluctuations. The average reading is compared with two example calibration values:

- DARK_LEVEL = 200
- BRIGHT_LEVEL = 800

If the sensor reading is greater than or equal to 800, the LED brightness is set to zero. If the reading is less than or equal to 200, the LED brightness is set to 255.For readings between these two limits, the Arduino calculates an intermediate PWM value using the map() function. The calculated brightness is applied to all three LED channels. The process repeats continuously.

### 6.2 Functional Operation

**Normal Daylight Condition:**
When the surrounding environment is bright, the LDR produces a higher analog reading. The Arduino sets the PWM value to zero, keeping all LEDs OFF.

**Low-Light Condition:**
When the surrounding light decreases, the LDR reading decreases. The Arduino increases the PWM value, gradually increasing LED brightness.

**Dark Condition:**
When the sensor reading reaches the predefined darkness limit, the PWM value becomes 255. All three LEDs operate at maximum brightness.

## VII. DATA ANALYSIS AND OBSERVATIONS

7.1 Sensor Reading Analysis

The Arduino Uno converts the analog voltage received from the LDR circuit into a digital value between 0 and 1023.The sensor reading changes according to the intensity of light falling on the LDR.In the proposed voltage-divider arrangement, bright light produces a higher reading, while darkness produces a lower reading.The program uses two calibration limits to determine the required LED brightness.

### 7.2 Functional State Analysis

**Table II: Expected Functional Response**

| Environmental Condition | LDR Reading | PWM Value | LED Condition |
|---|---|---|---|
| Bright daylight | ≥800 | 0 | All OFF |
| Medium light | 200–800 | Variable | Adjustable brightness |
| Dark environment | ≤200 | 255 | Maximum brightness |

The values shown above represent the example calibration settings used in the Arduino program. Actual readings may vary according to the sensor, resistor tolerance, sensor placement, and environmental conditions.

## VIII. ENGINEERING ADVANTAGES

The proposed system provides several engineering advantages:

- **Automatic Operation:** The controller adjusts lighting according to ambient light, reducing the need for manual switching.
- **Variable Brightness:** PWM provides intermediate brightness levels instead of restricting the LEDs to only ON and OFF states.
- **Reduced Unnecessary Illumination:** The daylight-OFF and automatic dimming behaviour avoids full LED output when it is not required.
- **Adjustable Control:** Daylight thresholds, darkness limits, and maximum brightness can be changed through software.



© 2026, IJSREM 

- **Low Component Cost:** The design uses commonly available Arduino, LDR, LEDs, and resistors.
- **Modular Architecture:** The sensing, processing, and output sections can be modified independently.
- **Educational Application:** The project demonstrates analog sensing, digital processing, and PWM control.

## IX. SYSTEM LIMITATIONS

The proposed system has the following limitations:

- **Sensor Dependence:** The system depends on the correct operation and placement of the LDR sensor.
- **Calibration Requirements:** Fixed threshold values may not suit every environment and require adjustment.
- **External Light Interference:** Shadows and nearby artificial lights can affect the sensor reading.
- **Power Supply Requirements:** A suitable and stable power source is required for reliable operation.
- **Limited Output Capacity:** The prototype uses low-power LEDs and cannot directly control high-power garden lamps without additional driver circuitry.
- **Outdoor Protection:** The electronic components require a weatherproof enclosure for outdoor installation.
- **No Wireless Monitoring:** The present design does not include remote control or mobile monitoring.

The project is an educational lighting prototype and is not intended to replace professionally designed outdoor electrical installations.

## X. FUTURE RESEARCH DIRECTIONS

The Smart Garden Light system can be improved through the following developments:

- **Improved Sensor Calibration:** More accurate calibration can be performed using measured illumination levels.
- **Solar Power Integration:** Solar panels can be used to provide renewable energy to the lighting system.
- **Motion-Based Lighting:** PIR sensors can be added to increase brightness when movement is detected.
- **Wireless Communication:** Wi-Fi or Bluetooth modules can be integrated for remote monitoring and control.
- **Intelligent Brightness Control:** Advanced control methods can be developed to improve brightness adjustment.
- **Weatherproof Design:** A protective enclosure can improve the suitability of the system for outdoor environments.
- **High-Power LED Integration:** Suitable MOSFET or LED-driver circuits can be used to control larger garden lighting installations.

## XI. CONCLUSION

This paper presents a Smart Garden Light system using an LDR sensor and Arduino Uno. The system integrates ambient-light sensing, microcontroller-based processing, and PWM-controlled LED brightness adjustment. The LDR detects changes in surrounding illumination and sends a corresponding analog signal to the Arduino. Based on the sensor reading, the controller adjusts the brightness of red, green, and blue LEDs.Under bright daylight conditions, all three LEDs remain OFF. As natural light decreases, the LEDs gradually become brighter and provide illumination during dark conditions.

## ACKNOWLEDGMENT

The authors sincerely acknowledge the Department of Electronics Engineering at [College Name] for providing the necessary laboratory resources, technical facilities, and academic guidance that made this project possible. The authors also express their gratitude to the project guide, faculty members, and institution for their valuable support and encouragement throughout the development of the Smart Garden Light project.

## REFERENCES

[1] Arduino, "Arduino Uno Rev3: Official Board Documentation." Available: https://docs.arduino.cc/hardware/uno-rev3/

[2] Arduino, "Arduino Programming Language Reference." Available: https://docs.arduino.cc/language-reference/

[3] Gontla Tarun et al., "Movement Based Automatic Intensity Control of Street Light," 2024.

[4] Soham Dey et al., "Energy Efficient Smart Street Lighting System Using Arduino," 2026.

[5] M. Margolis, *Arduino Cookbook*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media.

[6] J. Boxall, *Arduino Workshop: A Hands-On Introduction with 65 Projects*. San Francisco, CA, USA: No Starch Press.

[7] LDR Sensor Datasheet and Technical Specifications.

[8] RGB LED Datasheet and Technical Specifications.



© 2026, IJSREM | https://ijsrem.com 