Journal of Education and Culture Studies
ISSN 2573-0401 (Print) ISSN 2573-041X (Online)
Vol. 10, No. 1, 2026
www.scholink.org/ojs/index.php/jecs

Original Paper

Internet of Things and Smart Thinking: Design and Implementation of an Intelligent Lighting Control System Based on Arduino and Android

Wanduo Wang

Xihua University, Chengdu, Sichuan, 610039, China

Received: February 2, 2026 Accepted: March 13, 2026 Online Published: March 19, 2026

doi:10.22158/jecs.v10n1p110 URL: http://dx.doi.org/10.22158/jecs.v10n1p110

Abstract

With the popularization of computer technology and the rapid development of internet technology, social life is gradually moving towards intelligence. Concepts that were once theoretical, such as facial recognition payment and smartphones, have become reality, greatly facilitating people's daily lives. In this context, Internet of Things (IoT) technology, as an important component of the new generation of information technology, is promoting the vision of "connectivity of everything" and making our lives more intelligent. Smart home is a typical application scenario of IoT technology. By networking and automatically managing household devices, it significantly enhances living comfort and convenience. However, a complete smart home system is currently costly for ordinary consumers, making widespread adoption difficult. Therefore, this paper focuses on a core subsystem of smart home systems—the intelligent lighting control system—and designs a low-cost solution based on the Arduino and Android platforms. This system allows users to remotely control the on/off status and brightness of home lights through a mobile application, while also featuring a smart night light function that provides temporary illumination for users at night. This paper elaborates on the system's requirements analysis, feasibility study, system design, and implementation process, and tests the system functions. The results show that the system has basically achieved the expected functions and has certain practical value and promotion significance.

Keywords

Networking, Computer, Internet of Things, Intelligent Lighting Control, Arduino

1. Topic Description

(I) Problem Statement

In modern life, smart homes are gradually becoming prevalent. Intelligent management systems can automatically adjust indoor temperature and provide ventilation, offering users a more comfortable living environment. However, given the current market situation, installing a complete smart home system is costly and exceeds the budget of many ordinary households. As the most frequently used part of daily life, the lighting system has the most urgent need for intelligent transformation, but whole-house intelligent lighting solutions are also expensive. Therefore, designing a functionally focused, low-cost lighting control system to meet the needs of daily intelligent lighting has significant practical importance. This system should possess basic remote control and intelligent response functions, allowing users to experience the convenience of smart lighting without incurring a heavy financial burden.

(II) Project Tasks

The main task of this project is to design and implement an intelligent lighting control system based on the IoT architecture. Specifically, the system needs to achieve the following functions: users can remotely control the on/off status of home lights through a mobile application and adjust the brightness as needed; the system should also have a smart night light function that automatically provides temporary illumination upon detecting user activity at night, preventing users from fumbling for switches in the dark. To achieve these functions, the project involves the design and development of three main parts: mobile application design, for providing the user interaction interface; Arduino application design, for implementing the control logic of the hardware devices; and WEB server design, for handling data communication between the mobile end and the hardware end. Through the collaborative work of these three parts, a complete IoT application system will be built.

(III) Project Significance

The implementation of this project has multiple significances. First, for users, this system can significantly enhance the convenience of operating lights, especially providing great convenience for the elderly with limited mobility, allowing them to control lights without getting up and improving living comfort. Second, through the system's intelligent lighting management function, lights can be automatically adjusted based on actual needs, avoiding unnecessary lighting waste, thereby saving household electricity expenses to a certain extent, which has the significance of energy conservation and environmental protection. Third, for the project participants, through the design and implementation of this project, they can gain an in-depth understanding of the overall architecture of IoT applications and become familiar with technologies related to Android development, embedded development, and network communication. Through team collaboration, they can fully experience the project development process, enhance practical skills and teamwork abilities, and accumulate valuable experience for future study and work.

2. Feasibility Analysis

(I) Technical Feasibility

From a technical perspective, this project needs to be developed in two parts: hardware and software. In terms of hardware technology, it mainly involves embedded system development techniques for writing the control logic for LED lights, achieving on/off control and brightness adjustment, while also handling sensor data acquisition and response. In terms of software technology, it mainly involves Android application development techniques, requiring the development of a mobile app with a user-friendly interface capable of sending control commands to the embedded system via a wireless network. The more complex part is network communication processing, which requires ensuring stable and reliable data transmission between the mobile end and the hardware end. Currently, Android development technology is very mature, with abundant development documentation and open-source resources for reference; Arduino embedded development also has a large community support and numerous code examples. Therefore, from a technical standpoint, the implementation of this project is feasible.

(II) Economic Feasibility

In terms of economic cost, the design expenses for this system are relatively low. The main costs are concentrated on the procurement of hardware devices, including the embedded development board, sensors, LED lights, and other basic components. The software part mainly uses open-source technologies and development tools, such as the Android Studio development environment and the Arduino IDE platform, requiring no additional software purchase costs. Specifically, the required hardware includes: Arduino UNO R3 development board, Arduino Ethernet Shield W5100, home router, several ordinary LED lights, and a light sensor. These hardware devices are readily available in the electronics market, and the total cost can be controlled within 500 RMB. Compared to smart lighting systems on the market that cost thousands of RMB, this solution has a significant cost advantage and high economic feasibility.

(III) Equipment Feasibility

In terms of equipment selection, all hardware devices required for this design are common and easily obtainable models on the market. The Arduino UNO R3 development board serves as the system's control core, running the embedded program and processing control logic. The Arduino Ethernet Shield W5100 provides network connectivity for the development board, enabling it to access the local area network and communicate with the mobile application. The router builds the local area network environment, connecting the phone and the development board. LED lights simulate actual lighting equipment. The light sensor detects ambient light intensity, providing data support for intelligent dimming and night light functions. In terms of software, the Windows 10 operating system is needed as the development environment, Arduino IDE needs to be installed for embedded program development, and Android Studio needs to be used for mobile application development, along with configuring the corresponding SDKs and development tools. Both hardware and software are currently mature technologies with easily accessible resources, fully meeting the equipment feasibility requirements for the project.

(IV) Personnel Feasibility

From a personnel allocation perspective, the project team members possess the technical capabilities required to complete this project. Team members have accumulated experience in several technical areas: some members are familiar with backend development techniques and can handle server-side program design and data interaction logic; some members have Android development experience and can independently complete the interface design and functional implementation of the mobile application; some members master embedded development techniques and can write Arduino programs and debug hardware circuits. Through reasonable division of labor and team collaboration, each member can fully utilize their technical strengths and jointly promote the implementation of the project. Therefore, from the perspective of personnel capabilities, implementing this project is entirely feasible.

3. Requirements Analysis

(I) Business Process Flow

The business process flow of this system mainly revolves around the data interaction between the mobile application and the hardware device. When a user needs to control the lights via phone, the operation process is as follows: the user opens the Android application, which initiates an HTTP request using HttpClient. At this point, the phone is required to be connected to the local area network via WiFi, residing on the same network as the Arduino device. Predefined control commands are encapsulated in JSON format within the request and sent to the application running on the Arduino. Upon receiving the request, the Arduino end parses the JSON data, extracts the command content, completes the operation of switching or dimming the LED lights according to the command, and returns the operation result to the mobile application. Additionally, when the user opens the application, the APP first initiates an HTTP request to query the real-time status of all current lights. Upon receiving the query request, the Arduino end returns the status information of all current LED lights in the system in the form of a JSON list. After receiving the data, the mobile application parses and displays it, enabling the user to understand the current lighting status, providing a basis for subsequent control decisions.

(II) Business Functional Requirements

Based on the analysis of user scenarios, this system needs to meet the following core business functional requirements:

First, security and user management functions. The system should have a user authentication module, capable of adding and managing user accounts. User identity verification is required before each use of the system to ensure all control commands originate from legitimate users, preventing unauthorized individuals from maliciously operating the home lights and ensuring system security. Considering the home usage scenario, the system should support multiple family member accounts for different users' convenience.

Second, manual control function. Due to potential design flaws or other unforeseen issues during software and hardware operation, the system might crash or the network connection might be interrupted, preventing users from controlling lights via the mobile application. Therefore, the system must retain traditional manual control interfaces, i.e., physical switches, ensuring users can directly control the lights under any circumstances, meeting the basic usability requirements of the system.

Third, intelligent light adjustment function. With seasonal changes and weather conditions, natural light intensity constantly varies. The system should be able to detect ambient light intensity in real-time through a light sensor and automatically adjust the LED light brightness according to preset logic, maintaining indoor illumination at a comfortable level while achieving power savings. For example, automatically supplementing lighting when it gets dark in the evening, and reducing brightness or turning off lights during the day when light is sufficient.

Fourth, timing switch function. Each family member has their own routine. The system should allow users to customize automatic on/off times for each lighting device. For example, users can set the bedroom lights to automatically turn off at 11:00 PM every night and the living room lights to automatically turn on at 7:00 AM every morning. The timing function not only conforms to user habits but also avoids power waste caused by forgetting to turn off lights.

Fifth, smart night light function. At night, if a user needs to get up, the system should be able to detect user activity via a Passive Infrared (PIR) sensor or sound sensor, automatically turning on designated lights for temporary illumination. The brightness should be set to a low level to avoid strong light stimulation affecting the ability to fall back asleep, and the lights should automatically turn off after the user leaves, providing convenience while considering energy saving.

(III) Business Data Requirements

This system is designed for general household usage scenarios, where data volume is relatively small, and requirements for data processing and storage are not high. The data that the system needs to store mainly includes user-related information, such as account names, passwords (recommended to be stored encrypted), and user-defined timing settings for lights. This part of the data requires persistent storage using a database. Common choices could be lightweight databases like MySQL or SQLite. For environmental data collected by sensors, such as light intensity or human detection signals, the system does not need long-term storage. It only needs to read them in real-time and respond accordingly. These data can be discarded after completing the current control decision and do not need to occupy storage space.

(IV) Business Performance Requirements

The system's performance directly impacts user experience, so certain performance requirements must be met. First, response time is the most important performance indicator. The entire process, from the user clicking the control button to the actual change in the light's state, should be as short as possible. Referring to webpage loading user experience, if the waiting time exceeds 10 seconds, users will experience significant impatience. Considering network latency and system processing time, the response time for this system should be maintained within 1 second. This requires us to optimize algorithms and code logic during programming, reduce unnecessary processing steps, and ensure commands can be transmitted and executed quickly.

Second, the system should support multi-user concurrent login and control. As the system is used in a family environment, it's impossible to designate a single lighting controller. Multiple family members should be able to use the system to control lights simultaneously. Therefore, the system needs multi-threading processing capability to handle control requests initiated by multiple users concurrently, ensuring no conflicts or response delays occur when multiple people operate at the same time.

(V) Business Interface Requirements

The system's interface requirements can be divided into two categories: user interfaces and system functional interfaces. Regarding user interfaces, the system needs to provide basic management interfaces for adding, deleting, modifying, and querying user information, as well as a user authentication interface for identity verification before users log into the system. Regarding system functional interfaces, these mainly involve interfaces related to lighting control, including a device status query interface for obtaining the current on/off status and brightness information of all lights, a command issuance interface for sending control commands to specified devices, and a manual operation interface corresponding to the logic of physical switches, ensuring users can still operate lights traditionally when automatic control fails.

4. System Design

(I) System Architecture Design

This system adopts the classic three-layer IoT architecture for design, including the perception layer, network layer, and application layer. The perception layer mainly consists of the Arduino development board, LED lights, and various sensors, responsible for collecting environmental data and executing control commands. The network layer is implemented via the Ethernet shield and router, responsible for transmitting data between the mobile application and the hardware device. The application layer is the Android mobile application, responsible for providing the user interaction interface, displaying device status, and receiving user commands. This layered architecture design makes the functions of each layer clear, facilitating development and maintenance.

(II) Hardware System Design

The hardware system uses the Arduino UNO R3 as the core controller. It connects to LED lights via digital output pins for on/off control; it uses PWM pins to connect to dimmable LEDs for brightness adjustment. The light sensor connects to an analog input pin for detecting ambient light intensity. The Passive Infrared (PIR) sensor connects to a digital input pin for detecting human activity at night, triggering the smart night light function. The Ethernet Shield W5100 stacks onto the Arduino main board, communicating with the main control chip via the SPI interface to provide network connectivity. All hardware devices are powered uniformly, forming a complete hardware system.

(III) Software System Design

The software system includes three parts: the Android client, the Arduino program, and the Web service interface. The Android client adopts the MVC pattern design: the View layer is responsible for interface display, the Controller layer handles user interaction logic, and the Model layer manages data. The Arduino program adopts a modular design, including a network communication module, command parsing module, device control module, and sensor reading module. The Web service interface adopts a RESTful style design, defining unified API interfaces and using JSON format for data transmission, ensuring effective communication between the mobile end and the hardware end.

(IV) Communication Protocol Design

To ensure effective communication between the system components, a unified communication protocol needs to be designed. This system adopts an HTTP-based communication method, defining the following main interfaces: GET /api/devices for obtaining the status of all devices; POST /api/devices/{id}/on for turning on a specific device; POST /api/devices/{id}/off for turning off a specific device; POST /api/devices/{id}/brightness for adjusting the brightness of a specific device. All request and response data are encapsulated in JSON format, facilitating parsing and processing across different platforms.

5. System Implementation

(I) Hardware Implementation

Hardware implementation proceeds according to the circuit design. The Arduino UNO R3 and W5100 shield are connected by stacking, requiring no additional wiring. LED lights are connected to digital pins through 220Ω current-limiting resistors, with the anode connected to the pin and the cathode to GND. The light sensor and a 10kΩ resistor form a voltage divider circuit, with the analog output connected to an analog input pin. The PIR sensor's output pin connects to a digital input pin, while VCC and GND connect to 5V and GND respectively. After completing the connections, the Arduino program is uploaded to the development board for functional testing.

(II) Android Client Implementation

The Android client is developed using Android Studio with the Java language. The application interface includes a login interface, a main interface, and device control interfaces. The login interface implements user authentication, verifying username and password. The main interface displays all lighting devices in a list, showing the current status of each device and providing on/off control buttons. Clicking a list item enters the device control interface, where brightness can be adjusted and timing tasks can be set. Network communication uses the OkHttp library to send HTTP requests and the Gson library to parse JSON data.

(III) Arduino End Implementation

The Arduino program is developed using the Arduino IDE with the C++ language. During initialization, the program configures pin modes, initializes the Ethernet module, obtains an IP address, and starts an HTTP server. In the main loop, the program continuously listens for network requests and calls the appropriate handler function upon receiving one. For device status query requests, the program reads the current status of all LEDs, encapsulates it in JSON format, and returns it. For control commands, the program parses the JSON data and controls the corresponding pin outputs according to the command content to switch or dim the lights. Simultaneously, the program continuously reads sensor data and automatically controls the lights based on preset logic to implement the smart night light function.

6. System Testing

(I) Test Environment

System testing is conducted in a simulated home environment. Hardware devices include the Arduino UNO R3 development board, W5100 shield, three LED lights, a light sensor, a PIR sensor, and a home router. The software environment includes an Android phone with version 10 or higher and a Windows 10 development machine. All devices are connected to the same local area network to ensure normal network communication.

(II) Functional Testing

Various system functions are tested: In the remote on/off control test, each LED light is controlled separately via the mobile application. All commands are executed correctly, with a response time under 1 second. In the brightness adjustment test, adjusting each light's brightness via the application results in the light brightness changing according to the command. In the status query test, opening the application correctly displays the current status of each light, consistent with the actual status. In the smart night light test, triggering the PIR sensor in a night environment causes the designated light to automatically turn on and then turn off after a 30-second delay. In the timing switch test, after setting timing tasks, the system automatically executes the switching operations at the set times. In the user authentication test, only authorized users can log in and operate the system; unauthorized users cannot use it.

(III) Performance Testing

In the continuous operation test, the system runs stably for 72 hours without crashing or response failures. In the multi-user concurrency test, three users operating simultaneously are handled normally by the system without significant delay. In the network anomaly test, after disconnecting the network, the physical switches still control the lights normally, meeting the manual control requirement.

7. Conclusion and Outlook

Conclusion

After a period of effort, this intelligent lighting control system has basically achieved the expected functions. The system has realized the core functions of remotely controlling light on/off status and brightness adjustment via a mobile application, and possesses automation features such as a smart night light and timing switches. The system cost is controlled within 500 RMB, meeting the low-cost design goal. Test results show that the system operates stably, responds promptly, and can meet daily household usage needs. Admittedly, the system also has some shortcomings; some functions are not yet fully stable, and the user interface design is relatively simple, leaving a gap compared to commercial products. However, overall, this project has successfully validated the basic architecture of an IoT application and achieved the purpose of learning and practice.

Outlook

Future improvements and expansions can be made in the following areas: First, add more sensors, such as temperature/humidity sensors and smoke sensors, to expand system functionality. Second, connect to a cloud platform to enable remote control, breaking through the limitation of the local area network. Third, optimize the user interface to enhance user experience. Fourth, introduce voice control functions to further improve convenience. Fifth, research more intelligent control algorithms that learn from user habits to achieve a truly smart home experience. Through these improvements, the system can be further perfected, moving towards practical application.

References

Baron Schwartz et al. (2009). *High Performance MySQL*. Beijing: Kaiming Publishing House.

Guo Lin. (2016). *First Line of Code Android* (2nd ed.). Beijing: Posts & Telecom Press, 2016.11.

Xiao Mage. (2019). *Spring Boot Programming Ideas*. Beijing: Publishing House of Electronics Industry.

Zhang Fei. (2019). *High-Performance Android Development Techniques*. Beijing: Beijing University of Aeronautics and Astronautics Press.