Smart Environment Monitoring and Control System Using STM32 Microcontroller

Hasan Syed Tabeh, Md Sajjadujjaman Sohan, *Member,* and Anjum Saifullah, *Member*

***Abstract*—This document explains the design and implementation of a smart environment monitoring and control system based on STM32 microcontroller. The purpose of this system is to detect changes in light intensity and automatically respond by means of an LED and a buzzer, and at the same time provide useful information to the user with a colorful LCD. The system comprises a light sensor, speaker, push buttons, two LEDs (red and green), and a TFT LCD screen. Moreover, it has a settings mode that is protected by a password and can be transmitted to a computer through a UART port.**

The project sheds light on the modular design of the system which makes it easier to understand, test, and modify the code. Important functions have been separately arranged such as reading the sensor data, updating the display, button presses, and other UART commands. Data are displayed on the LCD as real time light levels and system status in an easy to understand way. With the buttons or UART commands, users can set the light threshold level and turn the buzzer on or off.

Such a system is applicable for automation in homes, classrooms, or in small IoT devices where light monitoring and control is essential. The results show that the system works well under different light conditions and responds quickly to changes, making it a reliable and practical embedded solution.

***Index Terms*—STM32, LCD Display, Light Sensor, Embedded Systems, Buzzer Alert, Smart Monitoring**

I. Introduction

In recent times, Coordination and automation of processes have, sparked interest in Environmental Monitoring Systems (EMS). Observation of these parameters within domestic, professional, and industrial scopes helps in managing, monitoring, elevating, safety, comfort, as well as improving productivity. Most traditional monitoring systems are limited in user adaptation and interaction since they are built around predefined operational logic or cumbersome hardware.

A cornerstone in the development of advanced smart monitoring systems are Embedded Systems. Such systems utilize cost-effective, low power, and miniaturized solutions for gathering sensor data, locally processing information, and interfacing with users through GUI based systems. Modern microcontrollers can be consolidated with a variety of peripheral modules including but not limited to, sensors, actuators and displays, to create bespoke intelligent systems for a wide range of application.

This document outlines the construction of Smart Environment Monitoring and Control System (EwMCS) built around the STM32 microcontroller. In this system, the ambient light level is continuously monitored and processed by an analog light sensor while the output actuators, LED indicators, and buzzer are controlled by pre-set thresholds. To enhance user interactivity, the system includes a colorful TFT LCD display (ST7735), a UART terminal interface, and a secure, password-protected configuration mode accessible via onboard keys. These features make the system not only functional but also adaptable and user-friendly.

The primary objective of this project is to demonstrate a cost-effective and interactive smart monitoring system that emphasizes modular embedded software, real-time feedback, and ease of configuration. The system is designed to respond dynamically to changes in ambient conditions, notify users via audio-visual cues, and allow threshold modifications through UART commands or physical buttons.

The main contributions of this work include:

Development of a modular embedded software framework on STM32 using C

- Real-time light-based actuation with configurable thresholds
- A visually informative LCD user interface for status feedback
- Secure entry to configuration mode using a 3-key password system
- UART-based command interface for interaction and debugging

The remainder of this paper is organized as follows: Section II details the hardware and system design. Section III describes the software architecture and algorithms. Section IV presents the experimental setup. Section V discusses the results and observations, and Section VI concludes with future directions.

May 10, 2025

II. SYSTEM DESIGN

A. System Hardware and Functions

This smart monitoring system comprises different hardware components regulated by an STM32 microcontroller to provide automatic light feedback and user control. All of the components play a significant role in the achievement of the system's objectives, as discussed in detail below.

i). STM32 Microcontroller (STM32F103C8T6):

The core part of the system is the STM32F103C8T6, a 32-bit ARM Cortex-M3 microcontroller. It is selected based on a trade-off between performance, price, and availability of peripherals. It has internal ADCs for sensor values, USART

This project is completed under the supervision of Prof. Li Yunrui as part of the Embedded Systems course.

All authors are students of the Embedded Systems course, 2025.

for UART communication, and a number of GPIOs to connect LEDs, LCD, buzzer, and push buttons. Its low power consumption and large developer community support make it ideal for embedded automation applications like this one.

ii). Light Sensor (LDR):

A Light Dependent Resistor (LDR) is used to measure ambient light intensity. The LDR's resistance drops inversely with respect to light intensity. When a series resistor is used in connection across a voltage divider, output voltage changes in proportion to light intensity, and this is monitored by the STM32's ADC. The reading from the analog input is interpreted as percentage (0–100%) and is used to trigger visual and audible alarms.

iii). TFT LCD Display (1.8" ST7735):

With use of the ST7735 controller, the system supports a color TFT display with a 1.8 diagonal viewing screen. It provides constant feedback regarding the sensor readings, system state, and user inputs. Enhancements like displaying messages in colors (blue for mid, red for low) improve the overall message clarity. To maximize pin savings and for high refresh rate, the LCD is connected via SPI interface.

iv). LEDs and Buzzer:

Two indicator LEDs provide immediate visual feedback—LED0 (Red) for high light, LED1 (Green) for low light, and LED0 and LED1 both lighting up if levels are medium. The piezo buzzer is triggered when light readings go beyond a user-defined threshold, offering an auditory signal. Multimodal feedback offers usability in both silent and ambient environments.

v). Push Buttons (KEY0, KEY1, WKUP):

Three local input temporary push buttons are employed. For changing the threshold level, KEY0 and KEY1 are utilized in normal mode, while WKUP toggles the buzzer status. The buttons become a digital keypad in settings mode to enter a password. Such two-mode design avoids hardware complexity without compromising secure interaction.

Power Requirements:

The system works with a regulated 5V USB power source and provides 3.3V step-down regulation when needed. Logic levels of the STM32 board operates at 3.3V, while other parts like the LCD and the sensors operate in the range of 3.3-5V. A stable supply gives uninterrupted sensor readings, LCD display, and clean buzzer sounds. During switching activities, the use of decoupling capacitor will lower the steady state noise supply.

This design is an embedded system that is interactive, power-efficient, and compact. With the use of solid modules, intuitive display, and feedback controls, the system will remain strong and easy to use when working on real ambient light monitoring applications.

B. Development of Smart Environment Monitoring System

Smart environment monitoring system integrates several hardware modules into one responsive embedded application. This paragraph elaborate on the integration and the functioning coherence of all parts. Developing every increment as a modular subunit offers a high degree of dependability and directness while simplifies the development process.

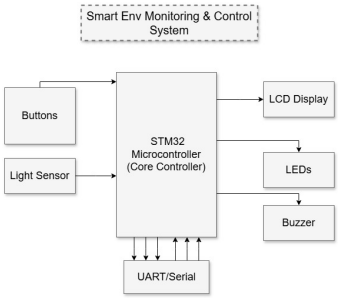

Fig. 1. Block Diagram

i). Module Interconnections

- The Light Sensor (LDR) is interfaced to one of the ADC pins (for instance, PA1). The sensor generates an electrical signal which is subsequently digitized.
- LCD (ST7735 TFT) is connected through SPI to PA5 (SCK), PA7 (MOSI), PA4 (CS), and DC, RESET for control lines.
- LED0 (Red) and LED1 (Green) are interfaced with GPIO (for instance PC13, PC14), permitting switching ON/OFF.
- The Buzzer is connected to another GPIO (PB8) that is used for sound generation.
- Two push buttons (KEY0, KEY1) along with WKUP are connected to GPIO pins configured with a pull-up resistor. Each button can be programmed for specific tasks based on the current operating mode.
- Communication with a PC terminal is achieved using UART (USART1) on PA9 (TX) and PA10 (RX). Connection is via USB or a serial-to-USB adapter.

All modules share a common ground, and regulated power (3.3V or 5V as required) is provided from the STM32 development board or USB supply.

ii). Module Characteristics Breakdown

- STM32: Obtains and processes sensor values, updates displays, manages input via UART, and executes logic.
- LDR: Determines light intensity and impacts the workings of the system.
- LCD: Displays the current readings and thresholds, as well as provide instructions to the operator.
- LEDs: Provides a visual indication of the prevailing light intensity (low, medium, high)
- Buzzer: Sounds alert when conditions meet threshold logic.
- Keys: Allows users to modify settings or input a password
- UART: Permits the user to command and supervise the system through serial communication via computer.

This division of responsibility improves modularity and simplifies debugging.

iii). Summary of Functions at the Same Level

- `display_lcd_status()`: Sets the LCD with the light value as system status and user hints dynamically.
- `check_light_and_control()`: Gets data from LDR and checks against thresholds to activate or deactivate LEDs and buzzer accordingly.

- `process_uart_command()`: Executes command, for instance setting thresholds and toggling the buzzer.
- `handle_keys()`: Detect key presses and change the system settings or enter a password.
- `main()`: Starts all modules and keeps monitoring and managing the system in an endless loop.

Each function handles a specific task and avoids unnecessary overlap, resulting in clean and efficient code.

v). Flowchart of System Behavior (Summary):

The system operate in a continuous loop where the STM32 microcontroller read the light sensor, evaluate the brightness level, and controls LEDs and the buzzer accordingly. The LCD display then updates with current readings and system status. User inputs via buttons or UART commands are processed for threshold adjustment, beep control, or password-based access. This flow ensures real-time monitoring, secure interaction, and automatic environmental response.

III. SOFTWARE DESIGN

A. Project Initialization Steps

The beginning of the Smart Environment Monitoring System software begins with basic hardware and peripheral initialization. During system booting, the STM32 microcontroller executes initialization routines for its clock, delay functions, UART communication, and all connected modules including LEDs, buzzer, LCD, light sensor (LDR), and input keys. Every module is initialize through its own initialization function to make the program systematic and readable.

When the initialization processes complete, a UART welcome message is sent and main system loop starts. The loop continually reads sensors and updates the LCD. Also, the loop is constantly checking if some button is pressed and processing UART commands. This system is designed to work forever, continuously reacting to changes the sensors and users make.

B. Modular Approach

In software, every module has almost the same functionality as a method in the object. In this case every function has its method, so the modules are divided in such a way that every function is self-contained. Some of the key functions are:

display_lcd_status():

The function is supposed to change the information shown on the LCD to match the actual state of things as they happen. Thus, the actual displayed information includes: current brightness level, threshold value, status of the buzzer, and system mode such as settings mode and password mode. The output is meant to respond adequately to user friendly interface and to the interaction between sensors and values inputs.

check_light_and_control():

This process checks light levels against specified thresholds set by the user. It controls LED0 (Red), LED1 (Green) and the Buzzer for these levels: LOW, MEDIUM and HIGH respectively. The Buzzer operates on a 0.5 second on/off timing aligned with the threshold being detected, further improving the responsiveness of the system.

process_uart_command():

This is dedicated to user commands transmitted via UART. Commands include reading sensor value, turning buzzer on/off, or setting thresholds and transitioning into settings mode. For protection, some of them are protect until a correct password has been typed from physical keys.

handle_keys():

This function considers the presses of physical keys (KEY0, KEY1, WKUP). The physical keys are used to type passwords, change thresholds, and turn on/off the buzzer when the system is in settings mode.

C. Password-Protected Settings Mode

In order to avoid accidental or unauthorize adjustments, a safe settings mode is implemented. It is accessible for the user by using UART with a special command. Once in it, the LCD prompts the user to type in a 3-digit password with the keys available (KEY0 = 1, KEY1 = 3, WKUP = 2). When the correct password (default: 132) is typed, the system permits access to adjust thresholds or change the buzzer. Unsuccessful passwords trigger an alert tone, and the system resets the attempt.

This mode increases system dependability in secure environments by restricting critical changes to privileged users only.

D. Light-Based Decision Logic

Intensity of light lies at the heart of system activity. System logic classifies light intensity in the following way:

- **Low Light (40%):** LED1 (Green) ON, buzzer beeps once.
- **Medium Light (40%–70%):** LED0 and LED1 ON.
- **High Light (70%):** LED0 (Red) ON.

The buzzer beeps only once for each state transition to prevent noise saturation. Such rules form a clever and responsive light-based monitoring logic.

E. Parsing UART Commands

The system can also be controlled through UART commands from a PC terminal. UART communication provides support for:

- `"0"`: Show help menu.
- `"1"`: Read the value of the light sensor.
- `"2"`: Turn buzzer ON/OFF.
- `"3 <value>"`: Set a new threshold for light.
- `"4"`: Enter settings mode (input password via keys).

The command handler employs string comparison to decode each command, and it offers an elegant means of handling user interactions.

F. Key Handling Logic

Three physical keys available for secure input are:

- **KEY0:** Taps digit 1
- **KEY1:** Taps digit 3
- **WKUP:** Taps digit 2

In settings mode, these keys are used for threshold increment/decrement and buzzer mode switching. Key input is debounced and read via the key_scan() function.

G. Flowchart Overview

The following is a simple overview of the software control flow:

- Start
- Initialize System
- Display LCD Status
- Read Light Sensor

- Compare with Threshold
- Update LEDs / Buzzer
- Validate UART Input
- Parse Commands
- Validate Keys
- Update System
- Loop Again

Each section is modular, easy to modify, and resource-efficient for use on embedded systems like STM32.

IV. EXPERIMENTAL SETUP

A. Hardware Used

The Smart Environment Monitoring and Control System was implement using the STM32F103 microcontroller as main controller unit. Hardware components integrated into the system include:

- **STM32F103C8T6 (Blue Pill)** board
- **TFT LCD display** (ST7735 module)
- **LDR** (Light Dependent Resistor) for sensing ambient light
- **Two LEDs** (Red and Green)
- **Active Buzzer** for alert sound
- **Push Buttons:** KEY0, KEY1, WKUP for user interaction
- **USB-to-Serial Converter** (for UART communication)
- **Breadboard and jumper wires**
- **3.3V power supply** (via USB or regulated adapter)

The STM32 board runs at 72 MHz, and the components are powered using 3.3V from the board itself. All modules are directly connected to the GPIO pins of the microcontroller.

B. Testing Scenarios To test the system's features, a number of practical test cases were created. Each scenario encapsulates some challenge the system could encounter during real life operations.

Case 1: Light Below Threshold When ambient lighting is below the predetermined threshold value, the green LED (LED1) is turned ON and the buzzer emits a short beep. The LCD displays "Light: LOW" together with the status of the buzzer. This case simulates a low-light scenario like nighttime or a dark room.

Case 2: Light Above Threshold For an extremely well illuminated room, above the specified threshold, the red LED (LED0) indicating high brightness gets illuminated. The LCD reads "Light: HIGH", and the buzzer remains OFF unless the light level decreases. This simulates daylight, or an environment that is too exposed.

A serial terminal such as Tera Term or PuTTY was used to monitor the UART interface through a computer. Commands "1" and "2" for light level and toggling the buzzer respectively were issued and successfully preformed.

Case 3: UART Instructions Through the interface, custom threshold modifications were achieved with command "3 ¡value¿", demonstrating real-time responsiveness.

Case 4: Handling Keypad Password Case Testing the password mode involved sending the command "4" through the UART interface. The LCD prompted the user to create a 3-digit password with the keys. Only after entering the correct password '132' did the system grant the user flexible settings adjustments. Incorrect attempts caused a reset and buzzer alert. This test confirms user control security.

C. Environmental Variables Aspects of ambient light for testing were simulated through hand shadows, flashlights, and room lights. An LDR quickly responded to stimuli, while the STM32 processes and respons to updates in milliseconds. The LCDs dynamic refresh performs without lag, with buzzer/LED reactions responsive.

The system operate as expect during testing, verifying the functionality and reliability of all modules within the test embedded smart environment system.

V. RESULTS AND DISCUSSION

The most important part of the visual feedback for the users was the LCD (ST7735) module. During the tests, the screen was able to show real-time value updates of the sensors, mode changes, and alert messages. For instance:

- `Light Level: LOW { LED1 ON`
- `Light Level: HIGH { LED0 ON`
- `Enter Password: ___`
- `Access Granted` or `Wrong Password`

Readability was ensured through the optimization of colors and font size, and dynamic message updating was achieved without screen flicker. It would help here to add a screenshot or photograph of the LCD with these messages.

B. UART Interface Screenshots

Through serial communication, another level of interaction could be established using the UART interface. These commands allowed users to control the system from a distance. The following outputs were capture using a serial terminal:

C. User Interaction Response Time

Every interaction with the user, whether it involved the buttons or the UART, was executed in close to real-time. The system architecture ensures responsive interfaces by continuously polling sensor values and user inputs every few milliseconds. The rate at which the LCD messages were updated and other hardware indicators such as LEDs or buzzers were activated was instantaneous.

Button presses were detected within a second, and UART commands had virtually instant feedback. This makes the system suitable for real-time monitoring and control.

D. Security via Password

Adding a layer of security with password protection greatly improved security. The settings-accessing default password was hardcoded to 132. The system would deny access and trigger a buzzer alert if an incorrect password was provided.

While the current password system is static and simple, it proved effective in restricting unauthorized access. The keypad method made the system more user-centric and intuitive.

E. Threshold Sensitivity Observations

The light threshold value could be adjusted via UART. Testing showed that the system reacted accurately to even small changes in ambient light. For instance, shifting the threshold from 100 to 90 made the LEDs and buzzer trigger sooner. This demonstrated that the LDR input and STM32 decision logic were precise.

However, external light changes such as shadows, reflections, or sudden light shifts could briefly trigger responses,

TABLE I
STRENGTHS AND LIMITATIONS OF THE SYSTEM

Strengths	Limitations
Real-time LCD feedback	Only one environmental parameter (light)
Responsive UART communication	Static (unchangeable) password
Easy user interaction via buttons	Small TFT screen display size
Buzzer/LED response to light changes	Requires wired UART interface
Modular, maintainable code structure	No remote/wireless control (e.g., Wi-Fi)
Portable and compact hardware setup	Limited expandability in current version

highlighting the importance of choosing an optimal threshold based on the environment.

VI. CONCLUSION

This project exemplifies the design and implementation of the Smart Environment Monitoring and Control System with STM32 microcontroller. The system sets an interactive framework that constantly monitors a certain environment light level and reacts accordingly with visual (LEDs), audible (buzzer), text (LCD) outputs, and capsule interactions through keys and UART communication.With multiple hardware modules comes a clear, modular software structure that enables the system to meet the responsive embedded application requirements.

This project has taught me the flexibility offered by modular design principles as well as what embedded systems offer. Each part such as the display, sensors, and control systems, and even the control systems can be designed with individual encapsulated as distinct functionalities which simplifies maintenance, extensibility, and debugging. The system's modularity alone increases its development efficiency and reliability while also making future adaptability easier.

The system can effectively fulfill requirements such as home automation which requires automatic lights, smart classroom that needs to adjust the lights or give alerts depending on the environment, and even early prototypes for an IoT based sensor solutions. Localized authority paired with user commands enables automation in confined areas to onboard intelligence.

As with every approach, there are limitations with this one as well. Contextual understanding is limited by the oversight of a single environmental sensor (light). While password protection is too rigid and inflexible granting security, it also makes mechanisms far too insensible. Its size and limited content still impede the graphical capability of an applying an LCD interface.

In implicating further the system's functionality, new integrations would have to be made to incorporate the existing system seamlessly. The integration of the ESP8266 or ESP32 as Wi-Fi Modules allows for IoT integration which includes remote access, cloud data logging, and more.

Designing a mobile companion app would provide even more control and improve user interaction. The addition of other sensors such as temperature, gas, humidity, or motion detectors would enhance the system's environmental awareness and its practicality in smart monitoring deployments.

In essence, this project serves as a starting point for constructing sophisticated and streamlined embedded monitoring systems. It requires few enhancements to be transformed into a well-rounded system and flexible monitoring platform for smart environments.

REFERENCES

[1] STMicroelectronics, "STM32F103 Reference Manual," RM0008, Rev 21, June 2020.
[2] Sitronix Technology Corporation, "ST7735R 1.8" TFT LCD Controller Datasheet," Version 1.2, 2010.
[3] M. A. Mazidi, S. Naimi, and S. Naimi, *STM32 Arm Programming for Embedded Systems*, 1st ed., Mazidi & Naimi, 2018.
[4] A. Bandyopadhyay, "Design and Implementation of Smart Monitoring Systems Using Embedded Controllers," *IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 2021, pp. 1–6.
[5] R. Kamal, *Embedded Systems: Architecture, Programming and Design*, McGraw Hill Education, 3rd ed., 2017.
[6] S. A. Sadiq and A. Ahmed, "Low-Cost Embedded Systems for Environmental Monitoring," in *Proc. IEEE International Conference on Green Tech (GreenTech)*, 2020, pp. 210–215.

APPENDIX

A. Source Code Snippet

Main control logic (simplified):

```
void check_light_and_control(void) {
    uint8_t light_val = lsens_get_val();
    if (light_val < threshold) {
        LED1(1);
        BEEP(1);
        delay_ms(500);
        BEEP(0);
    } else {
        LED0(1);
    }
}
```

Full code available on GitHub: github.com/tabeh5110/Smart-Env-Monitoring-Control-STM32.git

B. UART Command List

TABLE II
UART COMMAND SET AND FUNCTIONS

Command	Function
0	Show Menu
1	Read Light Sensor
2	Toggle Beep
3 <val>	Set Threshold (e.g., "3 60")
4	Enter Settings Mode (password needed)

ACKNOWLEDGMENT

The author gratefully acknowledges the guidance and support provided by the faculty and lab instructors of the Department of Electronics and Communication Engineering at Yulin University. Special thanks to Li Yunrui for their valuable suggestions and encouragement throughout the development of this STM32-based embedded system. Appreciation is also extended to the laboratory team for facilitating the required hardware resources during implementation and testing.