Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 22, No. 3, 2026

Design of a Smart Home Access Control System Based on ARM System

Jing Zhang, Chao Shen, Nuan Feng and Yuxun Zhou

Liaoning Institute of Science and Technology, Liaoning 117004, China

Abstract: With the rapid development of technology, smart homes are gradually becoming part of our daily lives, bringing lots of convenience. As a key device in home security, traditional locks, passwords, and IC card access systems have limitations like simple functionality and lower security. To address these issues, this paper designs and implements a smart home access control system based on the ARM-architecture STM32F103C8T6 microcontroller. It provides an efficient security management solution for smart home entry points and also offers technical references and ideas for optimizing the future adoption of embedded smart access control technology in both residential and commercial settings.

Keywords: Smart home access control; ARM architecture; STM32 microcontroller; multimodal authentication; remote control.

1. Research Background

Smart home access control is a core device of smart home security. It basically acts as your home's first line of defense and a key physical barrier. Most traditional systems just use keys, passwords, or simple IC card locks, which aren't very secure and are pretty basic to manage. They're easy to break and can't be controlled remotely. The access control system discussed here can fix these issues, giving users convenience without messing with safety. The plan is built around an ARM processor, taking both security and ease of use into account. These days, ARM tech for embedded systems, biometrics, and encrypted communication is getting solid and can support access control that uses multiple technologies. ARM processors use a RISC design, so they save power, run efficiently, and are easy to integrate—perfect for embedded and IoT products. This system combines smart recognition tech with an IoT platform, turning access control from just a physical lock into a full-on house security hub.

2. Hardware Design of the Smart Home Access Control System

The hardware design of this system is centered around the STM32F103C8T6 microcontroller. This chip, produced by STMicroelectronics, is a 32-bit ARM Cortex-M3 microcontroller based on the ARM Cortex-M3 core, with a maximum operating frequency of 72MHz. It has 64KB of Flash program memory and 20KB of SRAM for data storage, which meets the system's needs for data processing, program storage, and real-time control. It has abundant peripheral resources, integrating multiple communication interfaces, including USART, SPI, I²C, as well as ADC, DAC, general-purpose timers, and PWM control modules. This provides sufficient expansion capability for external sensors, display modules, and communication devices. It also supports various low-power modes, effectively extending battery life in power-supply scenarios. Connection options include USB, CAN, and Ethernet. The operating voltage is 3.3V, with a wide voltage tolerance, ensuring stable system operation.

For the minimal system design, the core circuit revolves around the STM32F103C8T6 and mainly includes the crystal oscillator circuit, power circuit, reset circuit, debugging interface, and external expansion interface.

The crystal oscillator circuit typically uses an 8MHz or 16MHz passive crystal oscillator as the clock source, providing a stable reference signal essential for CPU operation and peripheral synchronization. The power section uses a linear regulator to convert the external voltage to 3.3V, giving the microcontroller and related components clean, stable power. It also uses filter capacitors to control ripple and ensure power quality. The reset circuit, made up of a reset button, pull-up resistor, and decoupling capacitor, allows for hardware reset in case of system faults. The debugging/programming interface is designed using the SWD (Serial Wire Debug) method, requiring only two wires to handle program writing or online debugging, which simplifies development. Additionally, the minimal system reserves plenty of GPIO pins and communication interfaces, suitable for connecting various sensor modules (like recognition devices), actuators (like door controls and alarms), and displays, providing a flexible and robust hardware foundation for expanding system functionality.

The hardware design considered here follows the principles of "modular, expandable, and highly reliable." Each functional part has clear signal separation and electrical isolation, preventing mutual interference from affecting control quality. It also allows for functionality changes or scenario modifications without many restrictions. Overall, it can satisfactorily meet the technical requirements and stability expectations of smart access control systems for identity recognition, remote operation, and emergency alarms.

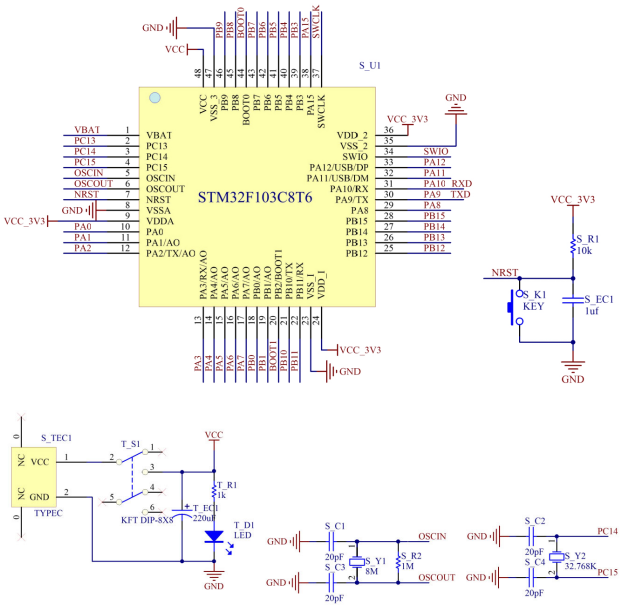


Figure 1. Microcontroller Minimum System Schematic

3. Software Design of the Smart Home Access Control System

This system is developed based on the STM32 series microcontrollers, using C as the main programming language. The C language, with its high level of abstraction, versatility, and concise, efficient code, allows direct manipulation of the underlying memory, making it an ideal choice for embedded system development. It effectively meets the system's strict requirements for real-time performance, reliability, and resource usage. The program development is done using the Keil uVision5 MDK integrated development environment, which provides comprehensive support for ARM-based processors, such as Cortex-M, Cortex-R4, ARM7, and ARM9 series microcontrollers. It integrates full-cycle functions like coding, compiling, linking, debugging, and downloading, enabling users to complete the entire software development process from design to implementation without switching tools.

In terms of software development, we first launch KeiluVision5MDK, create a new project, and precisely select the STM32 microcontroller model that matches the hardware platform to complete project initialization and configuration; Next, we write the core program code, covering modules such as system initialization, peripheral drivers, and functional logic implementation. After the code is completed, the "build target" function is used to perform compilation and linking. The system automatically detects syntax errors, logical conflicts, and resource usage to better resolve issues caused by writing. Developers need to check and fix all issues one by one until the compilation is free of warnings and errors. After compilation is successful, simulation debugging is initiated using "debug" mode. By setting breakpoints, stepping execution, and variable monitoring, we verify the correctness of program execution, logical judgment, and data processing, ensuring stable code operation in the simulation environment. After simulation verification is complete, the "Options for Target" option in the "Project" menu is configured to generate an Intel HEX format executable file, which contains machine code that can be directly programmed to the microcontroller's Flash memory. During the programming stage, strict operation guidelines must be followed: first disconnect the microcontroller's power, then use the downloader to flash the HEX file to the target hardware, restart the device, and start actual operation tests to avoid data transfer errors or hardware damage caused by powered operation. The system software adopts a modular design concept, dividing overall functions into four core modules, with clear interfaces interacting between modules to improve code maintainability and scalability. After the system is powered on, the first step is to run an initialization program, configuring key hardware such as the system clock, GPIO interface, UART serial communication, SPI/I2C peripherals, timer, and interrupt vector table, ensuring all functional modules are ready and functioning properly. Next, the main program enters a deadloop, where the four core functions are called sequentially to advance the subsequent run:

a. Key Handling Function: This function is divided into two subprocesses: key scanning and command execution. The key scanning subroutine periodically checks the status of physical keys, captures user input, and generates the corresponding key values. The command execution subprocess then triggers the corresponding actions based on the key values, such as switching interfaces, setting access passwords, starting fingerprint collection, or reading an RFID card. At the same time, the system actively reports the door lock's real-time status (such as locked/unlocked or error counts) to the mobile app via UART, allowing for status visualization. The mobile app can also control the lock's opening and closing remotely, enabling two-way interaction. UART1 refers to the first Universal Asynchronous Receiver/Transmitter in the microcontroller within an embedded system, representing a typical hardware serial port numbering. In embedded chips that support multiple UART peripherals, UART0, UART1, UART2, etc., correspond to separate hardware serial modules. All kinds of serial peripherals generally have similar basic functions and can independently configure parameters and operate functionalities, meeting the needs of multi-channel serial communication. Basically, a UART (Universal Asynchronous Receiver/Transmitter) is a widely used asynchronous serial full-duplex communication interface. This type of communication doesn't need a synchronous clock line and works just through the TX (transmit data) and RX (receive data) lines to enable point-to-point data transfer

between two devices. Thanks to its simple wiring and flexible configuration, it's a common solution for short-distance, low-speed serial communication in embedded systems.

b. Identity Verification Function: This function is a core module related to device security. It handles multi-modal security verification of user identities within the system, allowing fingerprints, numeric passwords, and RFID card IDs to be used individually or together, as needed. It can quickly retrieve the identity information provided by the user, denoise and correct the original signals, and eliminate noise, misreads, and data distortions, making the recognition result more accurate. The effective features obtained are then sent to the pre-stored Flash memory structure for legitimate credential comparison. The system itself has security mechanisms and log auditing set up. When an identity match is confirmed, device restrictions are lifted, history errors are cleared, and the current time, method, and user identity are recorded in a standard log format for secure keeping, making it easy for future analysis and maintenance. Errors triggered by failed identity matches are collected and indicated through buzzer and light alerts, directly pointing out anomalies. If consecutive errors reach a set security limit, it is considered a potential brute-force attack. At this point, verification paths should be closed, functions paused, and illegal use prevented. The function also safeguards against hardware, data, and time anomalies, ensuring verification results are not affected by malfunctions, forgeries, or prolonged downtime.

c. Lock Control Function: This function tracks all failed attempts during user authentication in real time, including failures from fingerprint recognition, password entry, and RFID card swipes. It keeps a unified tally, updates counts instantly, and dynamically evaluates errors, preventing resets from single failures or inaccuracies from improper categorization, ensuring correct counting. During normal processing of user information, whenever a mismatch, wrong password, or invalid RFID card occurs, the system immediately records the error count, gives basic audio-visual alerts, and guides the user to resubmit valid credentials. When the known errors reach 5 or more, the system treats the behavior as potential brute-force or malicious attacks and triggers proactive protection by immediately activating a forced lock. While locked, all authentication modules (fingerprint, keypad, RFID) are disabled, and identity verification operations cannot continue, cutting off any chance of repeated malicious attempts. The lock period is set to a fixed 3 minutes, with a countdown automatically controlled by the program. During this time, any requests are rejected and all actions blocked. Once the lock period ends, all restrictions are lifted, previous error counts are cleared, and all verification methods return to normal. This design provides clear protection against brute-force attacks and greatly enhances device security and authentication reliability.

d. Conflict Handling Function: The function we're talking about is a basic validation module in RFID card management. It fully collects and processes real-time raw data from RFID cards, then compares the valid card numbers obtained with the legitimate card whitelist stored in the system's Flash memory, checking everything thoroughly. For cases like adding new cards, modifying existing cards, or setting permissions, the function preemptively runs a conflict check, scanning all the card numbers currently in the system to accurately identify issues like duplicate numbers or data overlaps. If a new card number conflicts with an already confirmed valid card in the system, the function immediately stops the card writing or information storage process, refuses to accept duplicate entries, and ensures that original valid cards don't lose their permissions or stored status. At the same time, the system issues a status alert (like through indicator lights or a buzzer) to signal the operator to check the card, use an unregistered new card, or verify the existing card records. This detection method prevents overlapping card numbers, data confusion, and inconsistent permissions right from the data entry stage. It effectively avoids risks like system read/write errors, card permission failures, and device misjudgments caused by data conflicts, ensuring the uniqueness, completeness, and standardization of the card records, and greatly improving the stability and security of the RFID identity registration and permission system.

4. System Debugging

The physical test situation for the password unlocking function is shown in Figure 2. The test here is for checking the identity verification via keypad and the unlocking logic. The operator inputs the correct user password on the matrix keypad, and the microcontroller collects and analyzes the key values, sending them to the identity confirmation program for matching, comparing the entered password with the valid password credentials stored in Flash memory. Once confirmed correct, the relay starts working, the door lock indicator light turns on, the OLED display shows a prompt like 'Door is open' as agreed, and the buzzer emits a short beep, indicating that the unlock command was successfully executed. Now, considering the key processing, identity verification, driving part, and OLED display as part of the process, the system correctly identifies legitimate passwords, and all statuses appear clearly and immediately on the screen without lag or errors, achieving password unlocking as designed.

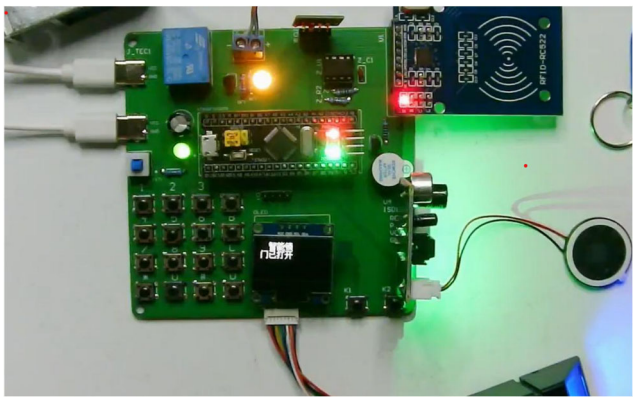


Figure 2. Correct Password Physical Test

The physical test results for the fingerprint entry function are shown in Figure 3. The main goal was to examine the reliability of the fingerprint module and the system entry process. When the system switches to fingerprint addition mode, the OLED screen displays the prompt 'Please press fingerprint,' instructing the user to place their finger on the fingerprint sensor area. The main control chip starts the registration program according to the button logic, and the sensor captures the user's fingerprint biometrics, preprocesses the relevant signals, and stores them in the designated storage. In terms of physical performance, the module can be correctly driven by the system, the on-screen instructions are timely and effective, and all hardware indicator lights turn on and off in sync with the collection steps. The tests conducted have

shown that fingerprint collection, feature analysis, and registration can all be successfully completed, interactive prompts are in place, and new user registration can be fully accomplished, thus providing a solid data foundation for subsequent fingerprint unlocking.

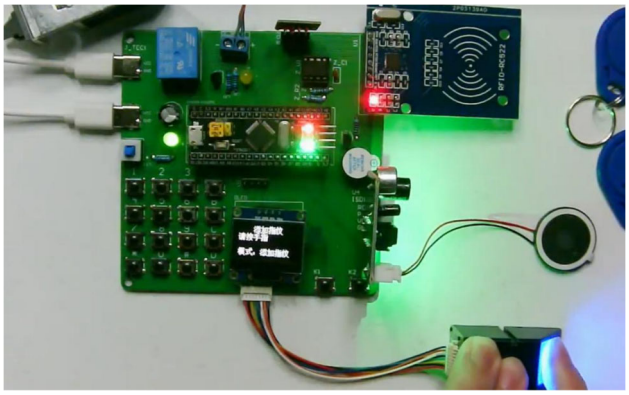


Figure 3. Fingerprint Input Physical Test

The physical test for card swiping to unlock with RFID, as shown in Figure 4, is designed to verify ID card recognition, number comparison, and unlock output. Placing a registered RFID card close to the RC522-type RFID reader module allows the reader to quickly obtain the card ID data and send the relevant information to the main STM32 microcontroller. The system then applies conflict handling and authentication procedures, comparing the retrieved card number with the whitelist of valid cards stored in Flash. When the card number matches, the relay activates, the door lock simulates unlocking, the indicator light turns on, and the OLED display shows 'Door Opened.' All card reading cases encountered during testing were timely and reliable, without any recognition delays or false triggers.

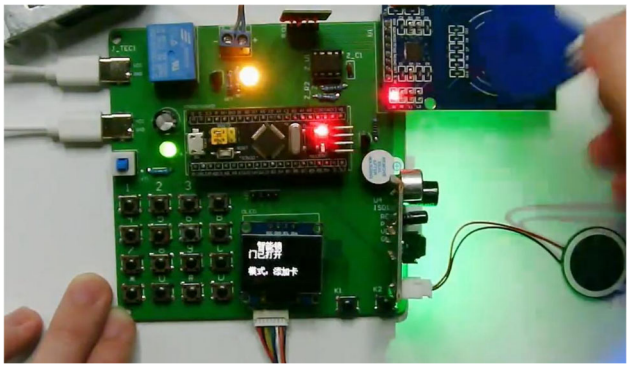


Figure 4. Physical Test of RFID Card Unlocking

The physical test for Bluetooth remote unlocking is shown in Figure 5. The purpose is to verify Bluetooth communication and remote commands. A smartphone connects wirelessly with the access control device via Bluetooth, and the user sends an unlock command to the system through a mobile app. The command is transmitted from the Bluetooth module to the STM32 main control chip via the UART serial port. The microcontroller analyzes the received command and logic, then operates the door lock; the relay responds accordingly, the onboard indicator lights change, and the OLED screen instantly displays the door lock status. From the test results, Bluetooth communication works well, the commands reach successfully, the system can recognize commands and control the lock, and the lock's status is fed back via the serial port. Therefore, the phone and access device have two-way communication, remote operation is feasible, and the access control method isget supplemented



Figure 5. Physical Test of Bluetooth-Controlled Lock Switch

From the practical perspective of this study, the smart access control solution mentioned in the paper basically meets all the functional requirements as well as security and reliability standards. However, there is still room for improvement in terms of intelligence, expandability, and practical usability. In the future, system development or upgrades should be considered from four aspects:

First, improvements in network connectivity, that is, turning WiFi or 4G into modules for IoT access control systems, so users can monitor, control, and review information remotely via their phones, freeing Bluetooth from range limitations.

Second, diversifying identity recognition, including 3D facial recognition, voiceprints, and various biometric features, creating a multimodal authentication system to further enhance the security and convenience of the access system.

Third, enhancing system storage and computing performance by upgrading chips (including main control and storage chips), increasing the storage capacity for data like fingerprints or RFID cards, serving more users simultaneously, and using AI or machine learning to track user behavior and predict abnormal situations, improving overall user experience.

Fourth, developing scenario-based adaptability by considering the different needs of home, office, or public spaces, creating customized functional modules, and integrating access control with building intercoms, smart security, smart home systems, and so on.

Overall, this paper provides a fairly complete discussion of the technologies involved in smart access control. The proposed solution has the potential to be a starting point for future development, allowing long-term improvements in functionality, usability, and scenarios, and can well meet users' expectations for convenience, security, and intelligence, thereby promoting wider use of embedded access control products in both residential and commercial applications.

5. Conclusion

This study uses the STM32F103C8T6 microcontroller as the core control unit and completes the design and implementation of a smart access control system that combines multiple functions, high security, and high-frequency usability. It provides a relatively complete analysis of the practical application of embedded technology in access control security and reasonably addresses the current societal

demand for diverse and intelligent access control systems. The system proposed in this paper centers on the microcontroller and incorporates fingerprint recognition, password input, Bluetooth communication, IC RFID card reading, and OLED display into the hardware design, while also developing the corresponding software logic to form a complete and coherent access control loop. From a functionality perspective, the unlocking methods include the four basic types: fingerprint, password, Bluetooth, and IC RFID card, greatly enhancing user convenience. In terms of permissions, the system allows simultaneous registration of 50 fingerprints or 10 RFID cards, with fingerprint and card data modifiable online, making the setup and adjustment of access permissions flexible. System security is considered through real-time monitoring of access status and alert arrangements, which can quickly detect unauthorized entry or device malfunctions and provide notifications. For human-computer interaction, the OLED display provides real-time feedback on the access status, doorbell triggers, and operation voice prompts, optimizing the user experience. Password management is categorized into administrator, fixed, temporary, and periodic types, meeting the different permission needs of residential and office environments. The data storage solution used is the AT24C02 type, which protects user data and configuration parameters from power loss, ensuring stable system operation and long-term data availability. Remote control is assisted by the Bluetooth module, allowing mobile terminals such as smartphones to issue commands and unlock doors, providing a supplemental usage scenario. The AT24C02 is a 2K-bit (256-byte) serial CMOS EEPROM, made with low-power CMOS technology, featuring a 16-byte page write buffer and supporting repeated erasing and data retention during power loss.

Acknowledgements

Home Access Control System of Yunshu Intelligent Box 202611430113

References

[1] Liang, H., Chen, J., & Chen, Y. (2021). Design of a smart home control system based on embedded ARM architecture. Journal of Guangxi University (Natural Science Edition), 46(1), 144–149.

[2] Sun, Y., Ye, N., & Kuang, Y. (2022). Design and implementation of a smart home control system based on the internet of things. Science & Technology Innovation New & Applications, 12(34), 110–113.

[3] Jiang, Z., Wang, W., Liu, S., et al. (2023). Design and implementation of a ARM-based smart home system. Modern Electronics Sub-technology, 46(4), 177–181.

[4] Wang, Y. (2022). Communication design of home security systems. Gansu Science and Technology Review, 51(4), 1–4.

[5] Liu, Y., & Hong, Y. (2024). Design of a smart home monitoring system based on ARM. Wireless Internet Technology, 21(6), 35–37.