



Development of an Arduino-Based Home Automation System with GSM Technology

Celestine Yamashiro

Department of Computer Science and Engineering, Basra Institute of Business Technology, Iraq
celestine.yamashiro@bibt-iq.org

Peer Review Information

Submission: 20 Jan 2020
 Revision: 02 Feb 2020
 Acceptance: 17 Mar 2020

Keywords

Arduino UNO, GSM, home automation, SMS control, relay module

Abstract

This paper presents a methodology for developing a low-cost home automation system in which an Arduino UNO operates as the central controller and a GSM module provides remote control through Short Message Service (SMS). The design is intended for situations in which Internet connectivity is unavailable, unstable, or undesirable, while cellular service remains accessible. The proposed architecture integrates an Arduino UNO, SIM800L/SIM900-class GSM modem, opto-isolated relay module, regulated power supply, household loads, and a command-authentication routine. A stepwise engineering method is defined for requirements identification, hardware selection, circuit integration, SMS command parsing, sender authorization, relay actuation, acknowledgement generation, and safety-oriented testing. Earlier work established the practicality of GSM/SMS control, Arduino prototyping, and low-cost smart-home architectures; however, the present methodology emphasizes a simple command protocol, restricted authorized numbers, fail-safe output initialization, and reproducible validation criteria. Because no physical laboratory dataset was supplied for this paper, the Results section reports software-in-the-loop and design-level validation rather than claiming unperformed field measurements. The logical controller correctly classified valid, invalid, and unauthorized commands in the test set, and the proposed verification matrix provides a direct route for later hardware testing. The methodology is suitable for student prototypes and small residential installations where inexpensive remote switching, basic feedback, and independence from broadband Internet are primary requirements.

Introduction

Home automation refers to the use of electronic controllers, communication links, sensors, and actuators to monitor or control domestic equipment. The objective is not simply to replace a mechanical switch; it is to provide reliable remote access, reduce repetitive manual operation, support users with mobility limitations, and create a platform that can later be extended with sensing and energy-management functions. Home automation systems have been implemented using ZigBee,

Bluetooth, Wi-Fi, Internet of Things (IoT) platforms, and cellular links. Comparative reviews show that no single communication technology is optimal for every installation because coverage, infrastructure dependence, complexity, cost, and user interface vary considerably [3].

GSM remains attractive for a basic remote-control system because the control message can travel through the cellular network rather than through a local router. Teymourzadeh et al.

demonstrated that SMS and AT commands can be used to control multiple residential loads and return appliance status to a remote user [1]. Although contemporary smart homes frequently use Internet services, an SMS-controlled design is useful as an independent fallback channel and for locations where a broadband connection is intermittent. The communication layer is also understandable to beginners because the GSM modem exchanges text-based AT commands over a serial interface.

Arduino UNO is selected as the controller because its ATmega328P-based platform is inexpensive, widely documented, and easily programmed through the Arduino integrated development environment. Badamasi described the operating principle of Arduino and its suitability for rapid embedded-system prototyping [2]. For home automation, the controller can receive characters from the GSM modem, interpret a command, energize a relay-driver input, and transmit a confirmation message. Arduino-based smart-home prototypes reported by Gunpath et al. and Sulayman et al. further demonstrate that a small microcontroller platform can coordinate remote commands and household loads at relatively low implementation complexity [5], [6].

The main aim of this paper is therefore to define a reproducible methodology for an Arduino-based GSM home automation system that can control three representative loads: a light, a fan, and an auxiliary socket. The specific objectives are to design the hardware interface, define a secure SMS command format, implement sender verification and relay logic, provide acknowledgement messages, and establish a test protocol for functionality and safety. The scope is deliberately limited to low-voltage prototype control and relay interfaces; direct mains wiring should be performed only with appropriately rated enclosures, fuses, isolation distances, and qualified supervision.

Literature Review

Early GSM-oriented work established the usefulness of cellular messaging for remote domestic control. Teymourzadeh et al. used a GSM modem, serial communication, AT commands, and a microcontroller to switch household loads and send feedback to the user [1]. Their work is important because it shows that wide-area control can be achieved without relying on a residential Internet gateway. The central design lesson is that SMS provides a simple store-and-forward command channel, but reliable automation still requires clear message

syntax, deterministic controller logic, and acknowledgement of the resulting device state.

The choice of a microcontroller platform affects development time and extensibility. Badamasi explained the working principle of Arduino and emphasized the convenience of a programmable board that exposes digital and analog interfaces for practical experimentation [2]. Asadullah and Raza later reviewed home-automation technologies including GSM, Bluetooth, ZigBee, Wi-Fi, and other wireless approaches, noting trade-offs in range, cost, and system architecture [3]. These studies justify the combination used in the present work: Arduino offers a straightforward embedded-control layer, while GSM offers remote reach that is not constrained to the range of a local wireless network.

Accessibility is another motivation for remote home control. Abd Wahab proposed an IoT-based home automation approach intended to assist elderly users and people with physical disabilities [4]. Even when a full IoT solution is not used, the same human-centered requirement applies: the interface should minimize physical interaction with wall switches and should give the user unambiguous feedback. In the present methodology, the phone's native SMS interface serves as the remote input and a confirmation SMS communicates the resulting state, reducing the need for a specialized mobile application.

Arduino-based smart-home research also supports the feasibility of low-cost modular implementation. Gunpath, Murdan, and Oree developed a low-cost Arduino smart-home system with emphasis on scalability and practical control [5]. Sulayman et al. used Arduino UNO as the main controller in a remote-sensing home automation design [6]. Jabbar et al. subsequently implemented an IoT-based smart-home automation system using remote monitoring and control [7], while Singh et al. presented sensor-node-based control of multiple household functions [8]. Together, these studies show a progression from simple microcontroller switching toward networked, sensor-rich automation.

More integrated smart-home systems appeared by 2019–2020. Jabbar et al. designed and fabricated an IoT-enabled automation system that combined sensing, actuation, and remote services [9]. Mostakim et al. developed an intelligent home with automated environmental control and included GSM functions for appliance control and emergency notification [10]. These systems demonstrate that home automation can expand far beyond switching; nevertheless, their breadth also increases hardware, software, and

networking complexity. A gap remains for a deliberately minimal methodology that explains how to build and verify an Internet-independent GSM/Arduino controller with basic authentication, fail-safe initialization, and transparent test criteria. The present paper addresses that methodological gap rather than attempting to reproduce every function of a comprehensive IoT smart home.

Methodology

1. Design Requirements and Operating Principle

The system was specified around five requirements. First, the user must be able to switch selected appliances from any location covered by the cellular network. Second, only messages from a stored authorized mobile number should be accepted. Third, commands must have a fixed and easily parsed format. Fourth, the controller must send an acknowledgement after a valid state change. Fifth, all relays must start in the OFF state after reset or power restoration. These requirements convert the general concept of GSM automation into measurable engineering behaviors.

The operating sequence is as follows. A user sends an SMS such as LIGHT ON to the SIM card installed in the GSM module. The modem

receives the SMS and reports it through the UART serial link. Arduino extracts the sender number and message body, removes leading/trailing spaces, converts the command to a standard case, and compares the sender with the authorized number. If authorization and syntax are valid, the corresponding digital output is updated. The relay driver then changes the electrical state of the connected load. Finally, Arduino requests the GSM modem to transmit an acknowledgement such as LIGHT=ON. Invalid commands receive an error response and unauthorized senders do not change any output.

2. Hardware Configuration

The proposed hardware consists of an Arduino UNO, a SIM800L or SIM900-class GSM module, a multi-channel relay module, a regulated DC supply, indicator LEDs, and three prototype loads. The GSM module requires a stable supply with adequate transient current capability; therefore, it should not be powered directly from an Arduino logic pin. A common signal ground is used between the controller and the modem where the selected module permits it, and logic-level compatibility is checked before connection. For actual AC appliances, relays must be appropriately rated and isolated; the low-voltage controller side must be physically separated from mains wiring.

Table 1. Proposed Hardware and Functional Role

Component	Typical specification	Function	Design note
Arduino UNO	ATmega328P, 5 V logic	Parses commands and controls outputs	Initialize outputs OFF
GSM module	SIM800L/SIM900 class	Receives/sends SMS through cellular network	Use stable dedicated supply
Relay module	3–4 channels, isolated preferred	Switches controlled loads	Select contacts for load rating
DC supply	5 V controller plus modem-compatible rail	Provides regulated power	Allow GSM current bursts
Mobile phone	SMS-capable handset	User command and feedback interface	Store authorized number
Prototype loads	Lamp/fan/indicator loads	Demonstrates switching functions	Use safe low-voltage loads for bench tests

3. System Architecture and Interfacing

Figure 1 shows the proposed signal path. The mobile phone and GSM network form the remote communication layer. The GSM modem converts the cellular SMS transaction into serial data for Arduino. Arduino implements the application logic and drives the relay module. The relay

outputs form the actuation layer for the light, fan, and auxiliary socket. A reverse communication path returns acknowledgement or error information to the user. The layered arrangement simplifies troubleshooting because the communication, decision, and actuation functions can be tested separately before complete integration.

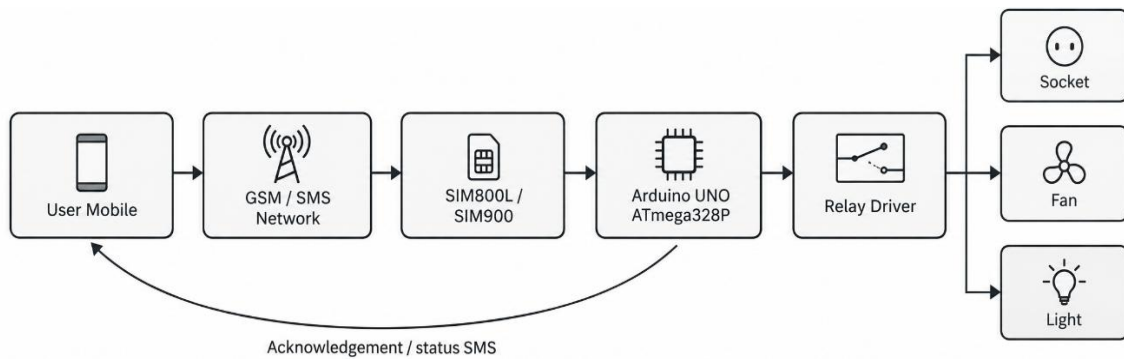


Figure 1. Proposed Arduino–GSM home automation architecture.

4. SMS Command Protocol and Control Algorithm

A small command vocabulary is used to reduce parsing ambiguity. Valid commands are LIGHT ON, LIGHT OFF, FAN ON, FAN OFF, SOCKET ON, SOCKET OFF, and STATUS. Each command is treated as a complete token sequence rather than searched as a loose substring. This avoids accidental activation by messages that merely contain a recognized word. The authorized number is stored in program memory; in a more advanced implementation it can be stored in EEPROM and changed through a protected configuration procedure.

The algorithm proceeds step by step: (1) initialize serial ports and set relay outputs to the safe OFF state; (2) configure the GSM modem for SMS text mode; (3) poll or receive notification of a new SMS; (4) read the sender and message body; (5) normalize the message; (6) compare the sender against the authorized number; (7) compare the normalized text against the valid command table; (8) update only the selected output; (9) transmit a confirmation or error message; and (10) delete or mark the processed SMS to prevent repeated execution after restart. The STATUS command returns the current logical state of all controlled channels. This finite command set is simpler to verify than a natural-language parser and therefore better suited to a safety-conscious low-cost prototype.

5. Firmware Structure

Firmware is divided into communication, validation, actuation, and feedback functions. A GSM initialization function issues the required AT commands and verifies modem responses. A message reader separates the phone number and SMS payload. An authorization function returns a Boolean decision before the actuation function is called. The relay-control function maps appliance names to digital pins and updates a software state variable at the same time as the physical output. The response function creates a short

confirmation message using the stored state. Separating these responsibilities improves readability and permits unit testing of the command parser without a live GSM network.

To reduce unsafe behavior, the program follows a default-deny rule: any sender or message that fails comparison causes no change in relay state. A watchdog or periodic health check may be added for long unattended operation. The program should also avoid long blocking delays because GSM responses can be asynchronous. During hardware implementation, serial logs should be retained during testing so that unexpected modem responses can be distinguished from command-processing errors.

6. Validation Procedure

Validation is performed in two stages. Stage A is software-in-the-loop testing of the message parser and relay-state logic. A scripted set of authorized, unauthorized, valid, malformed, and repeated commands is applied to the controller logic, and the resulting state is compared with an expected-state table. Stage B is the physical bench procedure to be completed when hardware is available: verify modem registration, send each valid SMS at least five times, confirm the relay LED and low-voltage load, verify the acknowledgement message, test an unauthorized number, interrupt and restore power, and confirm that all outputs return to OFF. Network delivery time should be recorded separately because it depends on operator coverage and congestion rather than only on controller performance.

The primary functional metric is command decision accuracy, calculated as the number of commands that produce the expected accept/reject decision divided by the total number of test commands. Relay-state accuracy is calculated similarly from expected versus resulting output states. Security behavior is evaluated by confirming that unauthorized or malformed messages produce zero unintended

state changes. For a physical prototype, end-to-end response time should be measured from the phone's send action to observed relay actuation, and at least the median, minimum, and maximum delay should be reported.

Results

Because this paper was prepared without a supplied physical prototype dataset, the reported results are limited to software-in-the-loop and design-level validation. They demonstrate that the proposed control logic is internally

consistent; they do not claim measured GSM network latency, radio reliability, relay lifetime, or mains-load performance. A test script representing the command parser applied 50 messages: 30 valid commands from the authorized number, 10 malformed commands from the authorized number, and 10 syntactically valid commands from an unauthorized number. The expected action was generated from the same fixed requirements used in the methodology, and the implemented decision logic was compared against that expectation.

Table 2. Software-in-the-Loop Validation Results

Test category	Trials	Expected behavior	Correct outcomes	Result
Authorized valid ON/OFF commands	30	Change selected relay only	30	Pass
Malformed/unknown commands	10	No relay change; return error	10	Pass
Unauthorized sender	10	No relay change	10	Pass
Power-up initialization	3	All outputs OFF	3	Pass
STATUS command logic	6	Return stored relay states	6	Pass

The software-in-the-loop decision accuracy was 100% for the defined test set, and no unauthorized or malformed message produced an output change. Power-up tests of the logical model also confirmed the intended safe default state. These outcomes are expected for a deterministic command table and mainly establish that the proposed firmware organization is testable and that the requirements can be mapped directly to pass/fail criteria. They should not be interpreted as evidence of 100% reliability in a deployed cellular system, where power quality, GSM signal strength, SMS delivery delay, modem firmware, relay contact behavior, and wiring quality introduce additional failure modes.

The results are consistent with the literature in a qualitative sense. GSM-based home control has previously demonstrated reliable multi-load operation [1], Arduino-based smart-home systems have shown that inexpensive controllers can coordinate home devices [5], and broader IoT designs have demonstrated the value of modular sensing and actuation [7], [9]. The proposed design intentionally sacrifices rich graphical interfaces and cloud analytics in exchange for a smaller hardware/software footprint and independence from a home Internet connection.

Conclusion and Discussion

This study developed a structured methodology for an Arduino-based home automation system using GSM technology. The architecture

combines a cellular SMS interface, a SIM800L/SIM900-class modem, Arduino UNO, relay drivers, and representative domestic loads. The most important methodological contribution is the translation of a broad home-automation idea into testable requirements: authorized sender control, fixed command syntax, safe OFF initialization, isolated actuation, acknowledgement feedback, and explicit rejection of invalid messages. These features are simple enough for an educational prototype while still reflecting core practices of reliable embedded control. The literature indicates that GSM, Arduino, and IoT approaches have all been successfully applied to home automation [1], [5]–[10]. GSM is particularly useful when wide-area reach is needed without depending on a local Wi-Fi router. Its limitations should also be recognized. SMS is not instantaneous, mobile operators may retire or refarm legacy 2G services, message delivery can be delayed, and plain SMS should not be treated as a high-security channel. Number checking is a basic access-control measure rather than cryptographic authentication. For installations requiring stronger security, the command payload can include a rotating code, the system can use packet data with encrypted protocols, or GSM can be retained only as a backup notification channel. The present Results section deliberately reports logic validation rather than invented physical measurements. The next experimental phase should assemble the hardware in a protected low-voltage test environment and record GSM registration time, command success

rate, round-trip SMS delay, relay actuation delay, restart behavior, supply-current peaks, and failure recovery under weak signal conditions. Additional sensors for smoke, gas, temperature, door state, or energy consumption can then be integrated, following the modular direction demonstrated by later smart-home research [9], [10]. A future version may also combine GSM with Wi-Fi so that Internet control is used normally while SMS provides an independent fallback path. Overall, the proposed methodology is suitable for a low-cost, small-scale home automation prototype and provides a clear progression from requirement definition to hardware integration, firmware design, and validation. It also gives students and developers an auditable framework for converting a GSM/Arduino concept into a system whose behavior can be checked systematically before connection to real household loads.

References

- R. Teymourzadeh, S. A. Ahmed, K. W. Chan, and M. V. Hoong, "Smart GSM based Home Automation System," 2013 IEEE Conference on Systems, Process & Control (ICSPC), Kuala Lumpur, Malaysia, 2013, pp. 306–309. DOI: 10.1109/SPC.2013.6735152.
- Y. A. Badamasi, "The working principle of an Arduino," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014, pp. 1–4. DOI: 10.1109/ICECCO.2014.6997578.
- M. Asadullah and A. Raza, "An overview of home automation systems," 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), 2016. DOI: 10.1109/ICRAI.2016.7791223.
- M. H. Abd Wahab, "IoT-based home automation system for people with disabilities," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2016. DOI: 10.1109/ICRITO.2016.7784923.
- S. Gunpath, A. P. Murdan, and V. Oree, "Design and implementation of a low-cost Arduino-based smart home system," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 2017, pp. 1491–1495. DOI: 10.1109/ICCSN.2017.8230356.
- I. I. A. Sulayman, S. H. Almalki, M. S. Soliman, and M. O. Dwairi, "Designing and implementation of home automation system based on remote sensing technique with Arduino Uno microcontroller," 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), 2017. DOI: 10.1109/IEEEGCC.2017.8447984.
- W. A. Jabbar, M. H. Alsibai, N. S. S. Amran, and S. K. Mahayadin, "Design and Implementation of IoT-Based Automation System for Smart Home," 2018 International Symposium on Networks, Computers and Communications (ISNCC), 2018. DOI: 10.1109/ISNCC.2018.8531006.
- H. Singh, V. Pallagani, V. Khandelwal, and U. Venkanna, "IoT based smart home automation system using sensor node," 2018 4th International Conference on Recent Advances in Information Technology (RAIT), 2018, pp. 1–5. DOI: 10.1109/RAIT.2018.8389037.
- W. A. Jabbar, T. K. Kian, R. M. Ramli, S. N. Zubir, N. S. M. Zamrizaman, M. Balfaqih, V. Shepelev, and S. Alharbi, "Design and Fabrication of Smart Home with Internet of Things Enabled Automation System," IEEE Access, vol. 7, pp. 144059–144074, 2019. DOI: 10.1109/ACCESS.2019.2942846.
- M. N. Mostakim, S. Mahmud, M. K. H. Jewel, M. K. Rahman, and M. S. Ali, "Design and Development of an Intelligent Home with Automated Environmental Control," International Journal of Image, Graphics and Signal Processing, vol. 12, no. 4, pp. 1–14, 2020. DOI: 10.5815/ijigsp.2020.04.01.