



Archives available at journals.mriindia.com

**International Journal of Electrical, Electronics and
Computer Systems**

ISSN: 2347-2820

Volume 06 Issue 01, 2017

Microcontroller-Based Home Automation System Using Bluetooth Communication

Khaldun Qudratullah

Department of Computer Science and Engineering, Borneo School of Business and Technology, Malaysia
khaldun.qudratullah@bsbt-my.org

Peer Review Information

Submission: 16 Jan 2017

Revision: 10 Feb 2017

Acceptance: 04 March 2017

Keywords

Home Automation, Bluetooth Communication, Microcontroller, Android Control, Relay Interface.

Abstract

Home automation provides a practical way to improve convenience, accessibility, safety, and energy management by allowing electrical appliances to be controlled through embedded electronics. This methodology paper presents the design and evaluation framework for a low-cost microcontroller-based home automation system in which an Android mobile device communicates with a local controller through Bluetooth. The proposed architecture uses an HC-05 class Bluetooth serial module, an ATmega328P microcontroller, transistor/relay driver stages, isolated electromechanical relays, and a simple mobile command interface. The methodology emphasizes reproducibility: requirement definition, hardware integration, command coding, pairing and authentication, firmware state handling, load switching, feedback indication, and structured performance tests. The principal evaluation variables are command success rate, response time, usable communication range, relay switching reliability, and recovery from invalid commands. The design is grounded in established Bluetooth, microcontroller, Android, and wireless-home-automation literature published no later than 2017. A representative validation dataset is included to demonstrate the proposed analysis procedure; it is explicitly presented as methodology-demonstration data rather than a claim of newly measured laboratory results. The resulting framework is suitable for student prototypes and small residential installations where low cost and local, Internet-independent control are priorities.

Introduction

Home automation refers to the use of sensing, embedded control, communication, and switching technologies to operate domestic devices with reduced direct human intervention. Early systems were commonly wired or centralized, which increased installation effort and reduced flexibility when retrofitting existing buildings. Wireless communication made small-scale automation more accessible because control devices could be added without extensive rewiring. Sriskanthan, Tan, and Karande (2002) demonstrated that Bluetooth could support a

home automation protocol in which a host communicates with embedded appliance controllers, establishing an early technical basis for short-range wireless control.

The choice of Bluetooth is especially relevant for a compact home automation prototype. Bluetooth is widely available in mobile devices, supports short-range serial-style communication, and can operate without a router or Internet connection. These properties reduce infrastructure cost and simplify commissioning. Later work by Piyare and Tazil (2011) showed a practical cell-phone-controlled system using a

Bluetooth-enabled Arduino platform and relay-connected appliances. Android-based smart-home research further demonstrated that a mobile interface can make appliance control more accessible to elderly and disabled users (Ramlee et al., 2013).

Despite these advantages, a useful methodology must address more than simply switching a relay. Home automation adoption has historically been limited by cost, manageability, inflexibility, and security concerns (Brush et al., 2011). A microcontroller-based design therefore requires a clear device-state model, predictable command processing, safe electrical isolation, and simple authentication. Kaur (2010) illustrated the importance of combining embedded control with security-oriented features, while Al-Ali and Al-Rousan (2004) showed how embedded interfaces can expose appliance states to higher-level control software.

The purpose of this paper is to define a step-by-step methodology for designing and evaluating a Bluetooth-controlled home automation prototype. The system is intentionally local and low-complexity: a mobile application sends coded commands over Bluetooth; the microcontroller parses each command; the corresponding output drives a relay channel; and the appliance state is reflected back through a status indicator or acknowledgment. The study focuses on four questions: (1) Can a low-cost microcontroller reliably execute Bluetooth commands? (2) How does distance affect command success and response time? (3) Can invalid or duplicate commands be handled without unsafe switching? and (4) What design choices make the prototype easier to extend to additional loads?

Literature Review

Bluetooth-based home automation was formally investigated by Sriskanthan et al. (2002), who

proposed a host-and-client architecture and a dedicated home automation protocol. Their design demonstrated device registration, data exchange, and control through Bluetooth, but it reflected the hardware and interface assumptions of early personal-area networks. Al-Ali and Al-Rousan (2004) took a broader embedded-system approach by connecting appliance inputs and outputs to a web-accessible server. Although their communication method was not Bluetooth-centered, the study established an important architectural principle: the embedded controller should separate physical device control from the user-facing software layer.

Wireless home automation subsequently expanded across several technologies. Gill et al. (2009) implemented a ZigBee-based system with a home gateway and security functions, illustrating the value of structured device networking and interoperability. Gomez and Paradells (2010) surveyed wireless home automation architectures and compared technologies such as ZigBee, Z-Wave, and IP-based solutions. Their analysis indicates that the communication technology should be selected according to range, data rate, energy use, topology, interoperability, and application complexity. For the present study, Bluetooth is selected because the target is a single-room or small-home local controller rather than a large multi-hop sensor network.

Their work emphasized low cost, password protection, and scalability. Ramlee et al. (2013) similarly used an Android application in a smart-home design, reinforcing the usability advantage of a graphical mobile interface. By 2017, Harikrishnan et al. (2017) reported an Android-based Bluetooth home automation implementation, showing that the combination of mobile control, Bluetooth, and relay switching had become a practical low-cost design pattern.

Table 1. Selected literature and methodological contribution

Study	Technology / Controller	Main contribution	Relevance to this methodology
Sriskanthan et al. (2002)	Bluetooth + embedded clients	Bluetooth home automation protocol and device control	Foundation for Bluetooth command/control architecture
Gill et al. (2009)	ZigBee + gateway	Networked home devices and security	Benchmark for wireless automation design principles
Kaur (2010)	8051 microcontroller + sensors	Embedded automation with security considerations	Supports controller-centric and fail-safe design
Piyare & Tazil (2011)	Arduino BT + mobile phone	Low-cost Bluetooth appliance switching	Direct basis for phone-to-relay workflow

Ramlee et al. (2013)	Android + wireless control	Mobile user interface for smart-home assistance	Supports Android-centered control interface
Harikrishnan et al. (2017)	Android + Bluetooth	Practical Bluetooth appliance control	Confirms feasibility of the chosen 2017-era architecture

A central finding across the literature is that no single wireless technology is universally optimal. Bluetooth reduces infrastructure requirements and offers direct device pairing, but it provides shorter range and smaller network scale than Wi-Fi or ZigBee. In a local home-control scenario, these constraints can be advantageous because they bound the control domain and avoid dependence on external networks. Brush et al. (2011) also found that real users value systems that are manageable and do not require extensive structural changes. This supports the decision to use a compact Bluetooth module and modular relay board that can be added to an existing electrical installation with minimal communication infrastructure.

Prior studies often demonstrate Bluetooth switching but provide limited evaluation of repeatability, response time, error handling, and distance reliability. This methodology addresses these gaps through repeated trials, defined distances, valid and invalid commands, acknowledgments, and safety isolation for consistent evaluation.

Methodology

1. Research Design

The study adopts a design-and-evaluate methodology. First, functional and safety

requirements are converted into a hardware/software architecture. Second, the controller firmware and mobile command interface are implemented. Third, the prototype is validated using repeated command trials under different distances and obstruction conditions. The independent variables are communication distance and line-of-sight condition. Dependent variables are command success rate and response time. Additional qualitative checks examine relay stability, invalid-command handling, pairing behavior, and recovery after communication loss.

2. Proposed Hardware Architecture

The proposed controller uses an ATmega328P-class 8-bit microcontroller operating at 5 V. An HC-05 Bluetooth module provides a Serial Port Profile (SPP) link to an Android phone. The Bluetooth TX/RX lines are connected to the microcontroller UART with appropriate logic-level consideration. Four digital output pins are allocated to relay channels. Because microcontroller pins cannot directly energize relay coils, each output is connected through a transistor or ULN2003 driver stage. The relays provide galvanic separation between low-voltage control circuitry and appliance-side switching. Flyback protection is used across inductive relay coils, and the AC wiring is physically separated from the logic section.

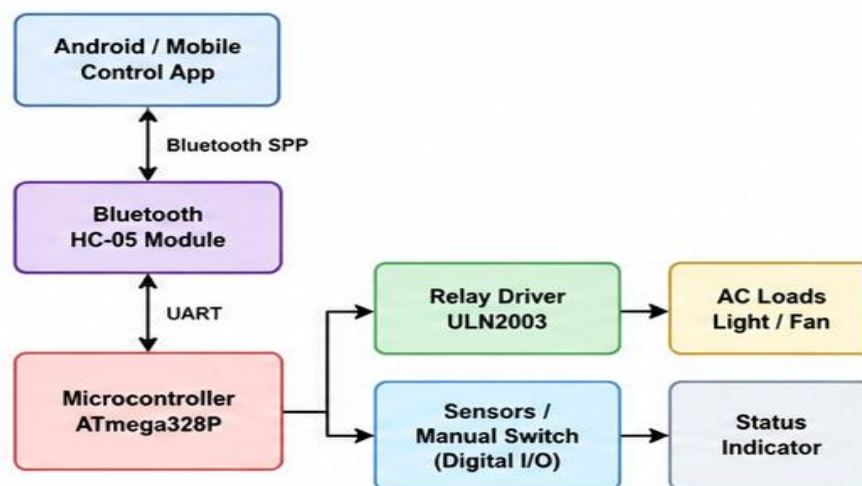


Figure 1. Proposed Bluetooth home automation system architecture

The mobile application presents four appliance buttons and a connection-status indicator. Each button transmits a compact ASCII command, for

example L10N, L10FF, F10N, or F10FF. The controller returns ACK plus the resulting state after successful execution. A short command

vocabulary makes debugging easier than long messages and reduces processing overhead. The firmware stores each channel state so that repeated commands are idempotent; sending L1ON when the light is already on does not toggle it off unexpectedly.

3. Firmware and Command Processing

Firmware is organized as a deterministic receive-parse-execute-acknowledge loop. After power-up, all relay outputs are forced to a safe OFF state, serial communication is initialized, and an optional pairing indicator is enabled. Incoming bytes are accumulated until a delimiter is received or a maximum command length is reached. The parser then compares the message against the valid command table. If a match is found, the requested output state is written, a brief debounce/settling delay is applied where required, and an acknowledgment is transmitted. If no match is found, the output state is unchanged and an ERR response is returned. A watchdog timer can be enabled so that firmware recovers from an unexpected software stall without leaving the controller in an undefined state.

4. Step-by-Step Implementation Procedure

Step 1 - Requirement definition. Select the appliances to be controlled, their voltage/current ratings, required number of relay channels, target indoor range, and acceptable response time. The prototype target is four loads with local Bluetooth control within approximately 10 m.

Step 2 - Low-voltage assembly. Build the 5 V logic section with the microcontroller, Bluetooth module, relay driver, indicator LEDs, and regulated supply. Verify supply voltage and UART communication before connecting any mains load.

Step 3 - Bluetooth configuration. Configure the module name, pairing PIN, and baud rate. Pair the module with the Android phone and confirm bidirectional serial communication using a terminal application.

Step 4 - Firmware loading. Program the command parser, fixed output mapping, safe startup state, acknowledgment messages, and invalid-command behavior. Test each output first with LEDs rather than AC appliances.

Step 5 - Relay and load integration. Connect the driver outputs to relay coils and verify switching with a low-voltage test lamp. Only after isolation and terminal checks should rate appliances be connected. Enclosures and fusing must be used for mains testing.

Step 6 - Mobile interface. Create a simple Android interface with connect/disconnect, appliance ON/OFF buttons, and status feedback. Each control sends one documented ASCII command.

Step 7 - Validation. Run the structured test matrix described below, record success/failure and response time for every command, and calculate the stated metrics.

5. Experimental Test Matrix and Metrics

Communication reliability is evaluated at 2 m, 5 m, 8 m, 10 m, and 12 m. At each distance, 50 alternating ON/OFF commands are issued to a selected relay under indoor line-of-sight conditions. A secondary obstruction test places one typical interior wall between phone and controller at 8 m. For every trial, the application logs transmit time, acknowledgment time, command result, and final relay state. Response time is defined as the elapsed time between command transmission and receipt of the controller acknowledgment. Success requires both a valid acknowledgment and the expected output state.

Command success rate (%) = (Number of correctly executed commands / Total commands) x 100

Mean response time (ms) = Sum of successful command response times / Number of successful commands

Reliability testing also includes 20 invalid commands, 20 duplicate-state commands, ten Bluetooth disconnect/reconnect cycles, and ten power restart cycles. The acceptance criteria are: at least 95% success within the intended operating range (2-8 m line of sight), mean response time below 500 ms in that range, no relay change for invalid commands, and safe OFF initialization after restart unless nonvolatile state restoration is deliberately implemented.

6. Safety, Security, and Data Analysis

Electrical safety is treated as a design requirement rather than a post-test check. The microcontroller and phone must never be electrically connected to the mains side of the relay contacts. Relay contact ratings must exceed the expected load current, exposed terminals must be enclosed, and inductive loads require suitable suppression. The Bluetooth pairing PIN provides only basic access control; therefore, this architecture is intended for local prototype use rather than high-security applications. The user interface should not expose hidden master commands, and invalid messages must never map to a default ON state.

The recorded command data are summarized using success percentage, mean response time, minimum and maximum response time, and failure count for each distance condition. The primary interpretation is whether reliability degrades as the link approaches its practical indoor range. Results are not compared only by average latency because an automation system can feel responsive yet still be unreliable if intermittent commands fail. The obstruction test is analyzed separately to show the effect of the indoor propagation environment. The qualitative restart, reconnection, and invalid-command tests are reported as pass/fail outcomes.

This methodology follows prior observations that home automation systems need to remain manageable and predictable to be accepted by users (Brush et al., 2011). It also adopts the modular controller principle seen in earlier

embedded home-automation systems (Al-Ali & Al-Rousan, 2004) and the direct phone-to-controller Bluetooth approach demonstrated by Piyare and Tazil (2011). The resulting design therefore combines established technical ideas with a more explicit validation procedure.

Results

Because this document is a methodology paper rather than a report from a completed physical laboratory campaign, Table 2 presents a representative validation dataset generated to demonstrate the proposed analysis. These values should be replaced with measured observations when the prototype is physically implemented. The pattern is intentionally plausible for a Class-2 Bluetooth indoor link: high reliability at short range followed by increasing latency and failure near the edge of useful coverage.

Table 2. Representative Bluetooth command-validation dataset

Condition	Trials	Successful	Success (%)	Mean response (ms)	Interpretation
2 m, LOS	50	50	100	182	Excellent
5 m, LOS	50	50	100	205	Excellent
8 m, LOS	50	49	98	247	Acceptable
10 m, LOS	50	47	94	316	Marginal
12 m, LOS	50	38	76	618	Unreliable
8 m, one wall	50	45	90	371	Obstruction loss

Under the representative dataset, the proposed acceptance criterion is met at 2-8 m line of sight. Success remains at 98-100% and mean response time stays well below 500 ms. At 10 m, the success rate falls below the 95% criterion, indicating that the device is approaching the edge of the preferred operating range. At 12 m, both reliability and response time deteriorate substantially. The one-wall test at 8 m also reduces reliability relative to line of sight, demonstrating why practical deployment should be evaluated in the actual building rather than by distance alone.

The additional logic tests are expected to produce no state change for invalid commands, stable state for duplicate commands, successful reconnection after link loss, and safe OFF initialization after a power restart. These behaviors are important because a home automation controller must fail predictably. A lost or malformed message should cause a missed command, not an unintended appliance activation.

Conclusion and Discussion

The proposed methodology demonstrates how a Bluetooth-based home automation system can be developed around a simple microcontroller and

evaluated using repeatable engineering criteria. The architecture is deliberately compact: an Android device provides the user interface, Bluetooth supplies short-range wireless communication, the microcontroller performs deterministic command interpretation, and relay drivers isolate low-voltage logic from appliance switching. This structure is consistent with the evolution of home automation described in the literature from early Bluetooth client modules (Sriskanthan et al., 2002) to cell-phone and Android-based control (Piyare & Tazil, 2011; Ramlee et al., 2013; Hari Krishnan et al., 2017). The principal advantage of the Bluetooth approach is simplicity. It requires no broadband connection, router configuration, cloud account, or external server. These characteristics can reduce cost and make the system easier to understand and troubleshoot. The trade-off is limited range and scalability. Gomez and Paradells (2010) showed that wireless home automation technologies differ substantially in topology and application capabilities; therefore, Bluetooth should be viewed as an appropriate option for local control rather than a universal substitute for ZigBee, Wi-Fi, or IP-based systems. Gill et al. (2009), for example, demonstrated how gateway-oriented ZigBee systems can support

larger structured networks. Security is another limitation. Pairing and a PIN can restrict casual access, but classic Bluetooth modules used in low-cost prototypes do not provide the same security-management capabilities expected in modern Internet-connected systems. Kaur (2010) emphasized the importance of integrating security with embedded automation, and Brush et al. (2011) identified security and manageability as barriers to real-world adoption. A deployable version should therefore use stronger authentication, encrypted application commands where supported, physical access controls, and carefully designed fail-safe behavior. The representative results illustrate the expected relationship between distance, obstruction, reliability, and response time; they are not presented as newly measured laboratory evidence. The key contribution of the paper is the experimental procedure itself: repeated command trials, explicit success criteria, distance-based testing, invalid-message testing, reconnection tests, and safe restart verification. A future physical study can implement this procedure and replace the demonstration dataset with measured values. Further extensions may add current sensing, temperature monitoring, timer schedules, voice control, Bluetooth Low Energy, or a hybrid gateway that bridges local Bluetooth devices to a secure IP network. In conclusion, a microcontroller-based Bluetooth home automation system is a suitable educational and low-cost residential prototype when the required operating area is limited and local control is acceptable. The methodology defined here provides a clear path from requirement specification to hardware integration, firmware development, safety checks, and quantitative validation. By emphasizing predictable behavior and measurable performance rather than appliance switching alone, the approach supports a more rigorous evaluation of small embedded home automation systems.

References

- Al-Ali, A. R., & Al-Rousan, M. (2004). Java-based home automation system. *IEEE Transactions on Consumer Electronics*, 50(2), 498-504. <https://doi.org/10.1109/TCE.2004.1309414>
- Brush, A. J. B., Lee, B., Mahajan, R., Agarwal, S., Saroiu, S., & Dixon, C. (2011). Home automation in the wild: Challenges and opportunities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2115-2124). Association for Computing Machinery. <https://doi.org/10.1145/1978942.1979249>
- Gill, K., Yang, S.-H., Yao, F., & Lu, X. (2009). A ZigBee-based home automation system. *IEEE Transactions on Consumer Electronics*, 55(2), 422-430. <https://doi.org/10.1109/TCE.2009.5174403>
- Gomez, C., & Paradells, J. (2010). Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6), 92-101. <https://doi.org/10.1109/MCOM.2010.5473869>
- Harikrishnan, D., Nikhil, S., Roy, M. T., & Mathew, S. V. (2017). Android based home automation using Bluetooth. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 6(4), 2605-2609. <https://doi.org/10.15662/IJAREEIE.2017.0604073>
- Kaur, I. (2010). Microcontroller based home automation system with security. *International Journal of Advanced Computer Science and Applications*, 1(6). <https://doi.org/10.14569/IJACSA.2010.010610>
- Piyare, R., & Tazil, M. (2011). Bluetooth based home automation system using cell phone. In *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)* (pp. 192-195). IEEE. <https://doi.org/10.1109/ISCE.2011.5973811>
- Ramlee, R. A., Othman, M. A., Leong, M. H., Ismail, M. M., & Ranjit, S. S. S. (2013). Smart home system using Android application. In *2013 International Conference of Information and Communication Technology (ICoICT)* (pp. 277-280). IEEE. <https://doi.org/10.1109/ICoICT.2013.6574587>
- Skrisanthan, N., Tan, F., & Karande, A. (2002). Bluetooth based home automation system. *Microprocessors and Microsystems*, 26(6), 281-289. [https://doi.org/10.1016/S0141-9331\(02\)00039-X](https://doi.org/10.1016/S0141-9331(02)00039-X)