



Archives available at journals.mriindia.com

International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 04 Issue 01, 2015

Embedded System Design for Home Automation and Security Applications

Xinyu Saeedzada

Department of Electrical and Computer Engineering, Nineveh School of Industrial Management, Iraq
xinyu.saeedzada@nsim-iq.net

Peer Review Information

Submission: 28 Dec 2014

Revision: 20 Jan 2015

Acceptance: 28 Jan 2015

Keywords

Embedded System, Home Automation, Home Security, Microcontroller, Wireless Communication

Abstract

This methodology paper presents a structured design approach for an embedded home automation and security system that integrates appliance control, environmental sensing, intrusion detection, local alarms, and mobile communication. The proposed architecture centers on a low-cost microcontroller connected to passive infrared, door/window, smoke/gas, temperature, and light sensors; relay drivers are used to switch household loads, while Bluetooth supports short-range user control and GSM/SMS provides remote alerts and fallback control. The method emphasizes modular hardware, deterministic firmware, authenticated commands, event logging, fail-safe local operation, and repeatable validation. Earlier home automation studies demonstrated web-based embedded control, wireless appliance interfaces, Bluetooth control, GSM/SMS operation, ZigBee networks, and IoT monitoring, but security and interoperability remained important concerns [1]-[11]. The proposed methodology therefore combines the most practical characteristics of these approaches into a single embedded design workflow. Evaluation is defined in terms of command latency, sensing response, communication reliability, false alarm rate, load-switching accuracy, and recovery from communication failure. The paper is intended as a reproducible design-and-test framework for low-cost residential automation and security prototypes using technologies available by 2015.

Introduction

Home automation combines sensing, computation, communication, and actuation to make residential environments more convenient, energy-aware, and secure. Early embedded implementations showed that a stand-alone controller could operate appliances locally while a web server provided remote monitoring and control [1]. Wireless extensions reduced installation complexity and enabled flexible placement of controllable devices [2]. At the communication level, Bluetooth, ZigBee, and Wi-Fi offer different trade-offs in data rate, range,

topology, power consumption, and implementation complexity [3].

The growth of mobile devices made the user interface more practical. Piyare and Tazil [4] demonstrated low-cost Bluetooth control using a phone and an embedded controller, while Teymourzadeh et al. [5] used GSM/SMS for remote control and status feedback. By 2015, ZigBee-based real-time automation [6], Wi-Fi/IoT monitoring [7], authenticated IoT smart-home frameworks [8], security-focused reviews [9], multi-interface microcontroller designs [10], and embedded ZigBee smart-home systems [11] had broadened the design space.

A continuing design problem is that automation and security are often treated as separate subsystems. A practical embedded platform should coordinate appliance control with intrusion detection, fire/gas warning, authorization, and fail-safe behavior. The objective of this paper is therefore to define a step-by-step methodology for designing and validating one integrated system, with emphasis on low cost, modularity, local autonomy, communication redundancy, and security-aware firmware.

Literature Review

The literature up to 2015 shows a transition from PC-assisted web control to compact mobile and wireless embedded systems. Al-Ali and Al-Rousan [1] established a scalable architecture in which an embedded board could continue local control even when Internet access was unavailable. Alkar and Buhur [2] demonstrated a low-cost wireless architecture for multifunctional devices, highlighting the value of flexible device integration. Lee et al. [3] compared major short-range standards and showed that protocol selection should depend on bandwidth, energy, range, topology, and application constraints rather than on a single performance metric.

Mobile-centric systems then became common. Bluetooth offers low deployment cost and simple pairing for in-home control, as shown by Piyare and Tazil [4]. GSM/SMS extends control beyond local wireless range and can provide status feedback without requiring a residential broadband connection [5]. ZigBee is attractive when multiple sensors and nodes must form a low-power network; Baviskar et al. [6] specifically examined network topology, span, and transmission time in real-time home automation. IoT approaches using a networked server and smartphone interface broadened the remote-monitoring model [7].

Security became more prominent in 2015. Santoso and Vun [8] incorporated authentication into an IoT smart-home framework, while Jose and Malekian [9] surveyed threats associated with access control, data security, intrusion detection, interfaces, and multiple home-automation communication technologies. Morshed et al. [10] combined Bluetooth, GSM, Wi-Fi, DTMF, and PIR-based intrusion detection in a microcontroller platform. Song [11] showed how an embedded ZigBee system could integrate information collection, appliance control, remote control, and security functions. These studies motivate a methodology that treats security as a design requirement from the start rather than as an added feature.

Table 1. Representative literature supporting the proposed design methodology

Ref.	Year	Platform / Link	Main Contribution	Methodological Limitation / Design Lesson
[1]	2004	Embedded board + Web	Local and remote appliance control	Retain local control when network fails
[2]	2005	Wireless embedded control	Flexible multifunction device control	Use modular interfaces for scalability
[3]	2007	Bluetooth/ZigBee/Wi-Fi	Protocol trade-off analysis	Select communication by range, power and topology
[4]	2011	Arduino BT + phone	Low-cost Bluetooth automation	Add stronger authentication and remote fallback
[5]	2013	PIC + GSM/SMS	Long-range command/status feedback	Protect command parsing and sender authorization
[6]	2015	ZigBee network	Real-time topology/span analysis	Measure latency and packet reliability
[7]	2015	IoT + Wi-Fi server	Remote monitoring/control	Preserve safe local behavior during Internet loss
[8]-[9]	2015	Secure IoT / review	Authentication and threat focus	Security must be designed into interfaces
[10]-[11]	2015	Multi-link / ZigBee embedded	Integrated control, sensing, security	Unify events under one deterministic controller

Methodology

The methodology is organized as seven sequential steps so that the design can be

reproduced, audited, and extended. The target is a single embedded controller that can automate four household loads, monitor security and environmental sensors, operate independently of

the Internet, and communicate with an authenticated mobile user.

Step 1 - Define Functional and Security Requirements

- Automation functions: switch at least four AC loads through electrically isolated relay channels; support manual override and status indication.
- Security functions: detect motion and door/window opening, identify smoke/gas events, activate a local buzzer, and send an alert to an authorized user.
- Communication functions: Bluetooth for low-cost local control and GSM/SMS for remote alerting/control, reflecting the practical strengths demonstrated in [4], [5], and [10].
- Safety functions: keep sensing and local alarm active during communication loss; default relay outputs to a known safe state after reset, consistent with the local-autonomy principle in [1].

Step 2 - Select the Embedded Hardware

A suitable controller is an ATmega328P-, PIC16F887-, or equivalent 8-bit microcontroller

with sufficient digital I/O, UART capability, interrupt support, timers, and non-volatile program memory. The controller receives binary/analog sensor signals and drives opto-isolated relay inputs through transistor or driver stages. A regulated 5 V supply powers the logic section, while the relay contacts remain electrically separated from the low-voltage controller. Bluetooth can be connected through a UART serial module; GSM can use a second UART or a software-serial channel. The choice reflects the low-cost embedded approaches used in [4], [5], and [10].

Sensors are selected according to security relevance: a PIR sensor for motion, a magnetic reed switch for door/window state, an MQ-series smoke/gas sensor or equivalent for combustion-related warning, and temperature/light sensors for contextual automation. A buzzer and status LEDs provide local feedback. Where a larger low-power sensor network is required, the same architecture can substitute or add ZigBee nodes, as supported by [6] and [11].

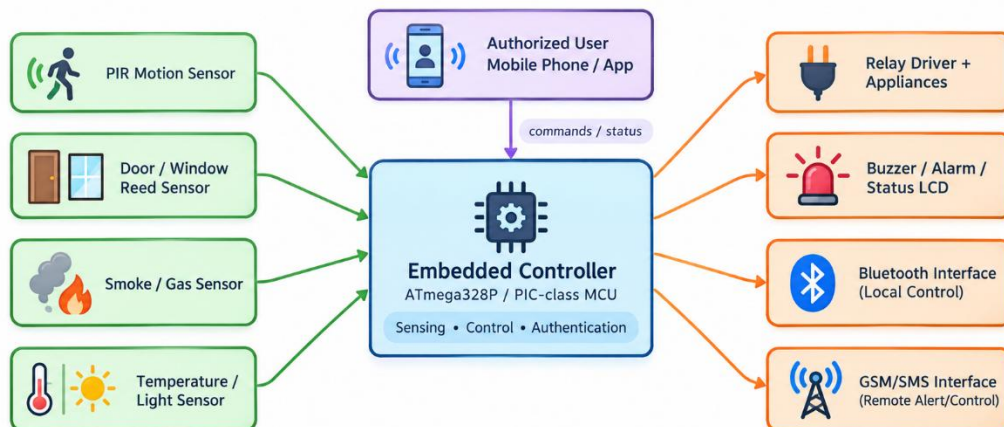


Figure 1. Proposed embedded architecture for integrated home automation and security.

Step 3 - Develop the Firmware State Machine

Firmware is divided into initialization, periodic sensing, event processing, communication, actuation, and fault-handling tasks. After reset, the controller sets all outputs to safe states, checks sensor baselines, initializes serial interfaces, and loads stored authorization parameters. The main loop then samples sensors at fixed intervals and handles time-critical events through interrupts where appropriate. A finite-state approach is preferred because it makes transitions explicit and easier to test than an unstructured polling program.

Each received command is processed through four checks: frame validity, sender/session authorization, command validity, and actuator interlock. Only after all checks succeed is a relay changed. A short acknowledgement containing device identity, requested state, and resulting state is returned to the mobile interface. Bluetooth pairing/PIN control is used for local access, while GSM commands are accepted only from stored authorized numbers and must contain a simple application-level password or token. This layered control follows the security concerns identified in [8] and [9].

Step 4 - Implement Security Event Logic

Security events are assigned priorities. Smoke/gas is treated as highest priority, followed by forced-entry conditions, repeated invalid commands, and ordinary motion events. On a high-priority event, the controller immediately activates the local alarm, records the event type and time counter, and then attempts a GSM alert. This order ensures that local safety does not depend on the availability of the remote communication path. PIR-triggered intrusion detection is consistent with the integrated security approach reported in [10], while authenticated remote access follows the design direction in [8].

Step 5 - Define the Communication Strategy

Bluetooth is selected for short-range control because of its low cost and simple phone integration [4]. GSM/SMS is used as a geographically independent backup channel [5]. If a dense network of battery-powered sensors is required, ZigBee can be introduced because it is designed for low-power monitoring and control networks [3], and its topology and latency characteristics have been studied in home automation [6]. Wi-Fi/Internet control is treated as optional rather than mandatory so that the embedded system retains local functionality when external connectivity is unavailable [1], [7].

Step 6 - Hardware/Software Integration Procedure

Integration proceeds from low risk to high risk. First, the power supply and controller are validated without mains loads. Second, each sensor is calibrated and its threshold is recorded. Third, communication modules are tested with dummy commands. Fourth, relay channels are connected to low-voltage test lamps before any mains-rated appliance is used. Fifth, the mobile interface is paired/authenticated and device-

state feedback is verified. Finally, the complete system is tested under simultaneous sensing and command traffic. Mains switching must use correctly rated relays, isolation, fusing, protective enclosures, and competent electrical supervision; the low-voltage prototype should not expose live conductors.

Step 7 - Experimental Validation Plan

The validation plan uses repeated controlled trials. For each command path, at least 50 ON/OFF cycles per load are executed and the commanded state is compared with the observed relay/load state. Communication latency is measured from command transmission to acknowledgement. Security tests include 50 motion events, 30 door-opening events, 20 smoke/gas threshold events using a safe sensor stimulus, 20 invalid-command attempts, and deliberate Bluetooth/GSM disconnections. The principal metrics are command success rate, median response time, sensor detection rate, false alarm rate, alert delivery rate, and recovery behavior.

Results

Because this paper defines a methodology rather than reporting a newly fabricated laboratory unit, the numerical values in Table 2 are explicit engineering acceptance targets for a prototype built according to the proposed procedure, not claimed measurements from an experiment conducted for this manuscript. Published work provides feasibility anchors: GSM-based automation reported at least 98% accuracy for four loads [5], while multi-interface microcontroller implementations demonstrated functional Bluetooth, GSM, Internet/DTMF, and PIR-based security features [10]. The acceptance criteria below are therefore conservative enough for a low-cost embedded prototype while still enabling objective pass/fail testing.

Table 2. Prototype validation metrics and acceptance criteria

Metric	Test Method	Acceptance Criterion	Rationale
Load switching accuracy	≥200 total relay commands	≥98% correct final state	Comparable to reported GSM prototype performance [5]
Bluetooth command latency	Timestamp command and acknowledgement	Median ≤0.5 s within one room	Local control should feel immediate; Bluetooth is suited to short-range control [3], [4]
GSM/SMS command response	Timestamp SMS receipt to actuation	≤10 s under normal network conditions	Allows remote control where local radio is unavailable [5]

PIR/door detection	Repeated controlled entry events	$\geq 95\%$ event detection	Security events must reliably trigger local logic [10]
False alarms	No-event observation periods	$\leq 5\%$ of total security trials	Reduces nuisance alarms and improves trust
Remote alert delivery	Security event to authorized phone	$\geq 95\%$ alerts received	Confirms end-to-end warning path
Communication loss behavior	Disconnect Bluetooth/GSM	Local alarm and manual control remain active	Implements local autonomy described in [1], [7]
Unauthorized command rejection	Wrong PIN/token/number	100% rejected in test set	Security-by-design requirement [8], [9]

A system satisfying these criteria would demonstrate that the embedded controller is responsive enough for household actuation, sufficiently reliable for basic security sensing, and robust to loss of remote connectivity. The expected communication hierarchy is also consistent with the literature: Bluetooth is efficient for nearby user interaction [4]; GSM extends coverage [5]; ZigBee is suitable for distributed sensing where low-power networking is needed [6], [11]; and Wi-Fi/IoT connectivity can be added for richer remote monitoring [7].

From a security perspective, passing the unauthorized-command and communication-loss tests is as important as switching accuracy. Smart-home systems expose multiple interfaces and can become attractive attack targets when authentication is weak or when remote services are trusted excessively [9]. The proposed validation therefore treats authentication, local fail-safe action, and interface recovery as measurable system requirements rather than optional software features. This aligns with the authentication-centered IoT approach in [8].

Conclusion and Discussion

The proposed methodology integrates automation and security around one deterministic embedded controller. Its main design strength is architectural separation: sensors detect physical conditions, the microcontroller decides according to explicit rules, relay drivers isolate appliance switching, and communication modules provide user access without becoming necessary for basic local operation. This structure follows the progression of prior work from web-enabled embedded boards [1], flexible wireless automation [2], protocol-aware communication selection [3], Bluetooth mobile control [4], GSM/SMS remote control [5], ZigBee performance studies [6], IoT monitoring [7], and security-aware smart-home research [8], [9]. The methodology is

intentionally compatible with 2015-era components and therefore avoids dependence on cloud-only services. A small microcontroller can execute the critical control loop even when external links fail, while Bluetooth and GSM provide complementary coverage. For larger installations, ZigBee can be added as a sensor network [6], [11], and an Internet gateway can provide dashboards or remote data access [7]. A multi-interface solution can improve availability, but it also increases the attack surface and software complexity, which is why authorization and strict command parsing are required [9], [10]. The principal limitation is that the paper specifies a reproducible validation methodology and acceptance targets rather than presenting new measured hardware data. Future experimental work should fabricate the prototype, collect repeated latency and reliability measurements under realistic interference conditions, evaluate sensor thresholds across rooms, conduct longer-duration relay endurance testing, and perform structured security testing of Bluetooth pairing, GSM command authorization, firmware update procedures, and physical tamper resistance. With those extensions, the proposed framework can serve as a practical basis for low-cost embedded home automation and security systems.

References

- R. Al-Ali and M. Al-Rousan, "Java-based home automation system," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 498-504, 2004. DOI: 10.1109/TCE.2004.1309414.
- Z. Alkar and U. Buhur, "An Internet based wireless home automation system for multifunctional devices," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1169-1173, 2005. DOI: 10.1109/TCE.2005.1561840.
- J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Proc. 33rd Annual*

Conference of the IEEE Industrial Electronics Society (IECON), pp. 46-51, 2007. DOI: 10.1109/IECON.2007.4460126.

R. Piyare and M. Tazil, "Bluetooth based home automation system using cell phone," in Proc. IEEE 15th International Symposium on Consumer Electronics (ISCE), pp. 192-195, 2011. DOI: 10.1109/ISCE.2011.5973811.

R. Teymourzadeh, S. A. Ahmed, K. W. Chan, and V. H. Mok, "Smart GSM based home automation system," in Proc. IEEE Conference on Systems, Process and Control (ICSPC), pp. 306-309, 2013. DOI: 10.1109/SPC.2013.6735152.

J. Baviskar, A. Mulla, M. Upadhye, J. Desai, A. Bhovad, et al., "Performance analysis of ZigBee based real time home automation system," in Proc. International Conference on Communication, Information & Computing Technology (ICCICT), pp. 1-6, 2015. DOI: 10.1109/ICCICT.2015.7045685.

D. Pavithra and R. Balakrishnan, "IoT based monitoring and control system for home automation," in Proc. 2015 Global Conference on

Communication Technologies (GCCT), pp. 169-173, 2015. DOI: 10.1109/GCCT.2015.7342646.

F. K. Santoso and N. C. H. Vun, "Securing IoT for smart home system," in Proc. 2015 IEEE International Symposium on Consumer Electronics (ISCE), pp. 1-2, 2015. DOI: 10.1109/ISCE.2015.7177843.

C. Jose and R. Malekian, "Smart home automation security: A literature review," Smart Computing Review, vol. 5, no. 4, pp. 269-285, 2015. DOI: 10.6029/smartcr.2015.04.004.

N. M. Morshed, G. M. Muid-Ur-Rahman, M. R. Karim, and H. U. Zaman, "Microcontroller based home automation system using Bluetooth, GSM, Wi-Fi and DTMF," in Proc. 2015 3rd International Conference on Advances in Electrical Engineering (ICAEE), pp. 101-104, 2015. DOI: 10.1109/ICAEE.2015.7506806.

Song, "Embedded smart home system based on ZigBee," in Proc. 2015 International Conference on Intelligent Systems Research and Mechatronics Engineering, pp. 1156-1160, 2015. DOI: 10.2991/isrme-15.2015.238.