Proceedings of CONF-MSS 2026 Syposium: Mechanical Control and Automation
DOI: 10.54254/2755-2721/2026.BJ32792

Application of Embedded System in Smart Home: Hardware Foundation of Sensor and Actuator

Yuyang Shi

School of Optical and Electronic Information, Suzhou City University, Suzhou, China
3211363160@qq.com

Abstract. With the rapid development of the Internet of Things (IoT) and microelectronics, smart home systems are becoming increasingly important. This paper examines the hardware foundations of embedded systems as applied to smart home environments, with a particular focus on the integration, operating principles, and practical deployment of sensors and actuators. Specifically, this paper investigates the technical characteristics of key hardware components including temperature and humidity sensors, smoke sensors, direct current (DC) motors, servo motors, and stepper motors. The study adopts a literature review methodology, systematically analyzing existing academic research, technical documentation, and engineering case studies to evaluate the roles of these components within embedded smart home platforms such as Arduino, STM32, and ESP32. This paper finds that embedded systems provide an efficient, low-cost, and highly adaptable hardware backbone for smart home applications, and that the effective integration of diverse sensors and actuators is central to enabling automation, energy efficiency, and user safety in residential environments. These findings contribute to a clearer understanding of the hardware design principles underlying modern smart home systems and offer reference value for future research and engineering practice in this domain.

Keywords: embedded systems, smart home, sensors, actuators

1. Introduction

The concept of the smart home refers to a residential environment in which various household devices and systems are interconnected and automated through digital technologies, enabling remote control, automated responses, and data-driven decision-making. At the core of these systems lies the embedded system — a specialized computing platform designed to perform dedicated functions within a larger device or system, typically under real-time operational constraints. Embedded systems are characterized by their compact form factor, low power consumption, and high reliability, making them ideally suited for deployment in domestic environments where continuous, unattended operation is required [1].

Research in smart home technology has progressed substantially over the past two decades. Early implementations focused primarily on remote control and rudimentary task automation, while more recent developments have incorporated machine learning algorithms, cloud connectivity, and

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.BJ32792

advanced sensor fusion techniques [2]. Despite this progress, the hardware foundations remain the indispensable physical layer upon which all higher-level functionality depends.

The motivation for this paper therefore arises from the growing practical importance of understanding hardware-level design principles as global smart home adoption continues to accelerate.

This paper expounds the working principle and application scenarios of core sensors and actuators in mainstream embedded platforms. The paper covers the definitions and distinguishing characteristics of embedded systems and smart homes, and analyzes the typical sensors and actuators used across common smart home scenarios. The study employs a literature review methodology. These findings are expected to support more informed hardware selection and system architecture decisions in future smart home development projects.

2. Embedded systems and smart home hardware applications

2.1. Definition and core characteristics of embedded systems

An embedded system is defined as a computer system with a dedicated function embedded within a larger mechanical or electrical system, often subject to real-time computing constraints [3]. Unlike general-purpose computers, which are designed for flexibility and multitasking across diverse applications, embedded systems are optimized to execute a specific set of tasks with maximum efficiency and reliability. Core characteristics include deterministic execution timing, robustness under harsh or variable operating conditions, and tight hardware-software integration.

Embedded systems typically consist of a microcontroller or microprocessor unit, memory modules (both volatile random-access memory and non-volatile flash storage), input/output interfaces, and peripheral communication buses such as I2C, SPI, and UART. These components collectively allow embedded systems to acquire data from external sensors, process it according to programmed logic, and drive actuators to produce desired physical outputs [4]. The ability to function autonomously with minimal human intervention is one of the most valuable attributes of embedded systems in the context of smart home automation.

Another important characteristic of embedded systems is their energy efficiency. In a smart home context, many embedded devices are required to operate continuously over extended periods, sometimes powered by batteries or energy-harvesting sources. Modern microcontrollers provide multiple low-power sleep modes that allow the system to minimize energy consumption during periods of inactivity while remaining capable of rapid wake-up in response to sensor events. This combination of continuous availability and low power draw is essential for sustainable smart home operation [5].

2.2. Basic concepts and technical architecture of smart homes

A smart home is a residence equipped with devices and systems that can be automatically controlled and remotely monitored through digital networks, enabling enhanced levels of comfort, security, energy efficiency, and overall convenience for occupants. The technical architecture of a typical smart home system is generally organized into three functional layers: the perception layer, the network layer, and the application layer [5].

The perception layer constitutes the physical interface between the smart home system and its environment. It comprises sensors that detect environmental parameters — such as temperature, humidity, light intensity, motion, and gas concentration — and actuators that carry out mechanical or

DOI: 10.54254/2755-2721/2026.BJ32792

electrical actions in response to control commands. This layer is directly managed by embedded microcontrollers, which handle real-time data acquisition and actuator control. The network layer includes the communication protocols and hardware infrastructure that enable data exchange between devices and central controllers. The application layer encompasses user-facing interfaces, cloud server infrastructure, and data analytics platforms that support remote monitoring, control, and automated decision-making [6].

Embedded systems serve as the central processing units of the perception layer, bridging raw physical signals from sensors and the digital communications required by the network layer. Their role is therefore foundational: without reliable embedded hardware performing accurate sensing and responsive actuation, the higher-level intelligence of a smart home system cannot be effectively realized.

2.3. Typical applications of sensors in smart homes

2.3.1. Application scenarios and operating principles of temperature and humidity sensors

Temperature and humidity sensors are among the most commonly deployed sensing devices in smart home systems, reflecting the central importance of indoor climate to occupant comfort and health. Their primary applications include Heating, Ventilation, and Air Conditioning (HVAC) automation, indoor climate monitoring, condensation prevention, and energy management optimization. By continuously measuring ambient temperature and relative humidity, the embedded system can automatically adjust air conditioning units, electric heaters, or ventilation fans to maintain predetermined comfort thresholds, often learning user preferences over time [7].

The DHT11 and DHT22 are two of the most widely used digital temperature and humidity sensors in embedded smart home projects. Both devices integrate a capacitive polymer humidity sensing element and a negative temperature coefficient (NTC) thermistor onto a single chip, outputting fully calibrated digital data via a single-wire serial communication interface. The DHT22 offers significantly higher measurement accuracy — ±0.5°C for temperature and ±2% to ±5% for relative humidity — compared to the DHT11, making it more appropriate for precision climate control applications [8].

Beyond comfort control, temperature and humidity data are also important for smart home energy management. Studies have shown that intelligent HVAC control driven by real-time sensor feedback can reduce residential energy consumption by a meaningful margin compared to conventional scheduled control systems [7]. This positions temperature and humidity sensing as not only a comfort feature but also a practical tool for sustainability in residential buildings.

2.3.2. Application scenarios and operating principles of smoke sensors

Smoke sensors serve a critical function in smart home safety systems, providing early detection of fire outbreaks or abnormal combustion events that could endanger occupants or property. The MQ-2 is one of the most frequently used gas and smoke sensors in embedded system projects, capable of detecting a range of hazardous gases and combustion byproducts including liquefied petroleum gas (LPG), methane, propane, hydrogen, and smoke particulates [9].

The MQ-2 sensor operates on the electrochemical principle of chemiresistivity: its active sensing element is composed of a tin dioxide (SnO_2) semiconductor material whose bulk electrical resistance decreases measurably in the presence of combustible gases or airborne smoke particles. Under normal air conditions, adsorbed oxygen on the surface of the SnO_2 grain boundaries creates a high-

DOI: 10.54254/2755-2721/2026.BJ32792

resistance barrier; when reducing gases or smoke particles are present, this barrier is diminished, causing a drop in overall resistance that can be detected by the microcontroller's analog-to-digital converter [10].

When the measured gas or smoke concentration exceeds a programmable threshold defined in the embedded control software, the microcontroller triggers a set of automated safety responses. These may include activating an audible alarm buzzer, transmitting push notifications to the homeowner's mobile device via the network layer, switching on exhaust ventilation fans, or automatically opening windows through motorized actuators [9]. This integrated response paradigm exemplifies the tight functional coupling between sensors and actuators that characterizes well-designed embedded smart home systems.

2.4. Applications of actuators in smart homes

2.4.1. Application scenarios and operating principles of DC motors

DC motors are among the most versatile and widely deployed actuators in smart home automation systems, finding application in scenarios requiring continuous rotary motion. Common use cases include motorized window curtain and blind systems, ventilation and exhaust fans, automated door and gate lock mechanisms, and conveyor-based household appliances. DC motors operate on the fundamental principle of electromagnetic induction: when a current-carrying conductor is placed within an external magnetic field, it experiences a mechanical force that, when applied to a rotating armature, produces continuous rotational motion [11].

In embedded smart home applications, DC motors are typically driven through dedicated H-bridge motor driver integrated circuits such as the L298N dual H-bridge module, which enable bidirectional control of both motor speed and rotation direction through pulse-width modulation (PWM) signals generated by the microcontroller.

2.4.2. Application scenarios and operating principles of servo motors

Servo motors constitute a specialized class of actuator that provides precise and repeatable angular positioning, distinguishing them from the continuous rotation characteristic of standard DC motors. This precision makes servo motors particularly well-suited for smart home applications requiring controlled angular displacement, such as smart door lock bolt actuation, orientation adjustment of solar-tracking panels, robotic arm manipulation in household assistance devices, and the control of adjustable ventilation louver angles [12].

A standard servo motor assembly incorporates three integrated subsystems: a DC motor providing the motive force, a gear reduction train that amplifies torque while reducing output shaft speed, and a potentiometer-based position feedback element connected to a built-in control circuit. The servo accepts a periodic PWM control signal with a standard period of 20 milliseconds; the width of the active pulse within this period — typically ranging from 0.5 ms to 2.5 ms — encodes the target angular position of the output shaft, generally across a range of 0° to 180°. The internal control circuit continuously compares the potentiometer-derived actual position to the commanded position and drives the motor to eliminate any positional error, implementing a closed-loop proportional control scheme [8].

Embedded platforms such as Arduino provide dedicated servo control libraries that abstract the precise PWM timing requirements, allowing developers to specify target angles directly in software without manual waveform programming. This simplicity, combined with the inherent positional

DOI: 10.54254/2755-2721/2026.BJ32792

accuracy and self-holding torque of servo motors, makes them a preferred actuator choice in many smart home embedded control applications.

2.4.3. Application scenarios and operating principles of stepper motors

Stepper motors are electromechanical transducers that convert discrete digital pulse signals into precise, incremental mechanical angular displacements, enabling accurate open-loop position control without the need for external position feedback sensors. This characteristic distinguishes stepper motors from servo motors and makes them advantageous in applications where feedback sensor integration is impractical or cost-prohibitive. In smart home environments, stepper motors find application in motorized window blinds and curtain rail systems, HVAC damper position controllers, automated furniture mechanisms, and precision-controlled valve actuators [13].

The operating principle of a stepper motor is based on the sequential energization of multiple stator coil windings, which creates a rotating magnetic field that the permanent-magnet or variable-reluctance rotor follows in discrete angular steps. The angular resolution of each step is determined by the motor's construction. The motor is typically driven by a dedicated driver integrated circuit, such as the ULN2003 Darlington transistor array, which provides sufficient current to energize the stator windings from the low-power output pins of a microcontroller. By precisely counting the number of pulses delivered to the driver, the embedded system achieves exact control of angular position and displacement without requiring any feedback element, greatly simplifying overall system design.

2.5. Integration of embedded systems with sensors and actuators

2.5.1. Applications of the Arduino platform

Arduino is an open-source embedded electronics platform comprising both a family of hardware microcontroller boards and a corresponding software development environment based on a simplified dialect of C/C++. First introduced in 2005 at the Interaction Design Institute Ivrea in Italy, Arduino has grown to become one of the most widely adopted platforms for prototyping, education, and deployment of smart home applications worldwide, owing to its exceptionally low cost, extensive and active community support, comprehensive peripheral compatibility, and minimal learning curve for new users [14].

The Arduino UNO board, which is based on the ATmega328P 8-bit AVR microcontroller running at 16 MHz, provides 14 digital general-purpose input/output (GPIO) pins — six of which support hardware PWM output — along with 6 analog input channels connected to a 10-bit analog-to-digital converter, and hardware peripheral support for UART serial communication, I2C, and SPI bus interfaces. This peripheral set is well-matched to the requirements of typical smart home sensing and actuation tasks [15]. The sensor interface is straightforward: the DHT22 temperature and humidity sensor connects to a single digital GPIO pin and communicates via a one-wire protocol handled by dedicated software libraries, while the MQ-2 smoke sensor produces an analog voltage output readable directly by the ADC channels. Servo motors are controlled through the Arduino Servo library, which manages the generation of correctly timed PWM waveforms transparently; DC and stepper motors are driven through external driver modules such as the L298N and ULN2003, controlled via standard digital GPIO and PWM outputs. The availability of thousands of open-source libraries covering nearly every conceivable peripheral significantly reduces development time and lowers the technical barrier for smart home prototyping.

DOI: 10.54254/2755-2721/2026.BJ32792

2.5.2. Other embedded platforms: STM32 and ESP32

While the Arduino platform excels in accessibility and ease of use for entry-level and prototyping smart home applications, systems with more demanding performance requirements, higher peripheral counts, or the need for integrated wireless connectivity typically benefit from more capable embedded platforms such as the STM32 microcontroller family and the ESP32 system-on-chip module [6].

The STM32 family, developed and manufactured by STMicroelectronics, is based on the ARM Cortex-M processor architecture and encompasses a broad spectrum of microcontrollers ranging from the entry-level Cortex-M0 devices operating at 48 MHz to high-performance Cortex-M7 variants with clock speeds exceeding 400 MHz. STM32 devices offer extensive hardware peripheral sets — including multiple UART, SPI, I2C, CAN, and USB interfaces, dedicated timer channels for high-resolution PWM generation, multi-channel 12-bit ADCs, and hardware floating-point computation units on higher-end variants. These capabilities render STM32 microcontrollers well-suited for smart home applications requiring real-time deterministic control, simultaneous management of multiple sensors and actuators, or computationally intensive signal processing tasks such as noise filtering and sensor data fusion [2]. The STM32 ecosystem is supported by professional development tools including the STM32CubeIDE and hardware abstraction layer (HAL) libraries provided by STMicroelectronics, which facilitate efficient peripheral configuration and reduce firmware development time.

The ESP32, developed by Espressif Systems and first released in 2016, integrates a dual-core Xtensa LX6 processor with a maximum clock speed of 240 MHz alongside native Wi-Fi 802.11 b/g/n and Bluetooth 4.2 (including BLE) wireless connectivity within a single low-cost module. This integration of processing power and wireless communication capability makes the ESP32 particularly well-suited for IoT-enabled smart home deployments, where sensor data must be reliably transmitted to cloud servers, remote dashboards, or mobile applications in real time [5]. The ESP32 supports lightweight IoT communication protocols such as Message Queuing Telemetry Transport, enabling efficient publish-subscribe data exchange with cloud platforms including Amazon Web Services IoT Core, Google Cloud IoT, and open-source home automation platforms such as Home Assistant. With its combination of processing performance, rich peripheral set, and wireless connectivity in a compact, energy-efficient, and low-cost package, the ESP32 has rapidly become one of the leading choices for modern IoT-connected smart home embedded applications.

The selection among Arduino, STM32, ESP32, and other embedded platforms ultimately depends on the specific requirements of the smart home application in question. For educational purposes or simple single-sensor prototypes, Arduino provides the lowest barrier to entry. For multi-sensor real-time control systems requiring deterministic timing, STM32 offers superior performance. For cloud-connected IoT deployments requiring wireless communication, ESP32 provides an optimally integrated solution. In practice, complex smart home installations may employ multiple platforms simultaneously, with each assigned to the tasks best suited to its capabilities [6].

3. Conclusion

This paper has systematically examined the hardware foundations of embedded systems as applied to smart home environments, with a focused analysis of the operating principles, application scenarios, and system integration strategies of core sensors and actuators.

The central conclusion of this paper is that embedded systems provide the indispensable physical and computational backbone for smart home automation, with sensors enabling accurate

DOI: 10.54254/2755-2721/2026.BJ32792

environmental perception and actuators enabling responsive, controllable physical action. The effective selection and integration of appropriate hardware components — matched to the specific functional requirements of the application — is a critical design consideration that directly determines system performance, reliability, energy efficiency, and cost-effectiveness. Among the platforms reviewed, Arduino offers the greatest accessibility and development simplicity for educational and prototyping contexts; STM32 provides superior computational performance and peripheral flexibility for real-time and processing-intensive applications; and ESP32 represents the most capable solution for IoT-connected deployments requiring embedded wireless communication.

This paper is subject to certain limitations. The analysis is based entirely on literature review methodology and does not incorporate experimental validation. Furthermore, future research in this field could profitably focus on the integration of on-device artificial intelligence and machine learning inference at the embedded hardware level, enabling edge computing capabilities that reduce latency and dependence on cloud processing. Moreover, achieving seamless interoperability and standardized communication across heterogeneous smart home devices from different manufacturers remains a persistent engineering challenge.

References

[1] Barr, M., Massa, A. Programming Embedded Systems: With C and GNU Development Tools [M]. O'Reilly Media, 2006: 1-20.
[2] Yiu, J. The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors [M]. Newnes, 2013: 15-30.
[3] Heath, S. Embedded Systems Design [M]. Newnes, 2003: 3-25.
[4] Predko, M. Programming and Customizing the PIC Microcontroller [M]. McGraw-Hill, 2008: 200-230.
[5] Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions [J]. Future Generation Computer Systems, 2013, 29(7): 1645-1660.
[6] Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M. Internet of Things for Smart Cities [J]. IEEE Internet of Things Journal, 2014, 1(1): 22-32.
[7] Zhou, B., Li, W., Chan, K. W., Cao, Y., Kuang, Y., Liu, X., Wang, X. Smart Home Energy Management Systems: Concept, Configurations, and Scheduling Strategies [J]. Renewable and Sustainable Energy Reviews, 2016, 61: 30-40.
[8] Monk, S. Programming Arduino: Getting Started with Sketches [M]. McGraw-Hill Education, 2016: 88-110.
[9] Behera, T. M., Mohapatra, S. K., Samal, U. C., Khan, M. S., Daneshmand, M., Gandomi, A. H. Residual Energy-Based Cluster-Head Selection in WSNs for IoT Application [J]. IEEE Internet of Things Journal, 2019, 6(3): 5132-5139.
[10] Postolache, O., Pereira, J. D., Girao, P. S. Smart Sensors Network for Air Quality Monitoring Applications [J]. IEEE Transactions on Instrumentation and Measurement, 2009, 58(9): 3253-3262.
[11] Chapman, S. J. Electric Machinery Fundamentals [M]. McGraw-Hill Education, 2011: 530-560.
[12] Acarnley, P. P. Stepping Motors: A Guide to Theory and Practice [M]. IET, 2002: 1-15.
[13] Kenjo, T., Sugawara, A. Stepping Motors and Their Microprocessor Controls [M]. Oxford University Press, 1994: 20-45.
[14] Banzi, M., Shiloh, M. Getting Started with Arduino [M]. Maker Media, 2014: 1-30.
[15] Margolis, M. Arduino Cookbook [M]. O'Reilly Media, 2011: 300-340.