American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN: 2320-0936
Volume-14, Issue-12, pp-40-46
www.ajer.org
Research Paper Open Access

The Chicken Nanny: An Autonomous Heating and Feeding Chicken Coop

Kuo-pao Yang[1], Patrick McDowell[2], Adam Rodi[3], Adriean Lemoine[4], Chris Smith[5], Jonathan Freeman[6]

[1,2,3,4,5,6]Department of Computer Science, Southeastern Louisiana University, Hammond, LA, USA
Corresponding Author: kuo-pao.yang@sotheastern.edu

ABSTRACT *This paper presents the design, implementation, and evaluation of an automated chicken coop system developed to optimize poultry farming by automating essential coop functions, including feeding and temperature regulation, designed for small-scale poultry farmers. This system employs an Arduino microcontroller programmed in C++ to regulate consistent feed flow through a servo-driven gate mechanism. A clock component enables user-defined feeding schedules. A relay component facilitates the safe management of a heating system, maintaining an optimal temperature to promote chicken health and productivity. The developed Chicken Nanny system successfully demonstrates the reliability and effectiveness of the automation, indicating its potential for broader agricultural applications.*

KEYWORDS *Arduino; Embedded Systems, Thermocouple Sensor, Automation.*

Date of Submission: 12-12-2025 Date of acceptance: 24-12-2025

I. INTRODUCTION

Poultry farming, whether on a small or industrial scale, demands significant daily management to ensure the health, productivity, and overall well-being of chickens. Critical tasks such as feeding and maintaining optimal temperature are traditionally performed manually, requiring substantial human effort and increasing the risk of inconsistencies and oversight. Manual management can lead to feeding irregularities and improper temperature control, negatively affecting poultry health, productivity, and profitability.

Existing solutions to automate poultry farming typically involve high-cost commercial systems or partial automation that still require considerable human oversight. These systems are often inaccessible to small-scale farmers due to their high initial investment and complexity. Additionally, many available solutions lack flexibility in scheduling and adaptability to various environmental conditions, limiting their effectiveness and broad adoption.

This proposed solution addresses these limitations by offering a fully integrated, cost-effective automated chicken coop that leverages an Arduino Uno microprocessor programmed in C++ to deliver precise, user-defined feeding schedules and temperature regulation. Integrating a real-time clock component allows customizable feeding schedules, while a servo-driven, 3D-printed gate mechanism ensures reliable feed distribution. A relay and a thermocouple sensor safely and accurately manage a 120-volt heating element, maintaining optimal temperature conditions with minimal human intervention.

This automated system is superior due to its affordability, simplicity, reliability, and adaptability, making it especially appealing to small-scale poultry farmers and hobbyists. By providing a low-cost yet highly effective automation solution, our system significantly reduces manual labor, improves consistency in care, and enhances overall poultry health and productivity.

The remainder of this paper is structured as follows. It provides a detailed overview of related work and existing automation approaches, presents the implementation methodology including architectural design, hardware integration, and software logic, evaluates the performance of the developed system with results from practical testing scenarios, and finally concludes by summarizing the findings, potential improvements, and future directions for research and development.

II. RELATED WORK

Automating poultry farming processes such as feeding and temperature management has been the subject of significant research across academia and industry. Prior systems span from basic servo-driven feeders to advanced Internet of Things (IoT)-based environmental monitors.

On the commercial side, industrial operations often deploy robotized feeding systems or sensor-driven automation. These solutions, while sophisticated, are costly and overly complex for small-scale farmers. Innovations in robotic animal husbandry have further explored such systems [1], emphasizing that modern solutions often require centralized control infrastructure and network dependencies, factors that the Chicken Nanny intentionally avoids.

Do-it-yourself (DIY) approaches based on Arduino or Raspberry Pi platforms are more accessible. For example, an Arduino-based chicken farm monitor integrating temperature and humidity sensors has been built [2][3], but it lacked mechanical feeding actuation and user configuration. Similarly, an IoT pet feeder has been developed [4], but it focused on remote web integration rather than environmental adaptability or heating.

Existing robotized feeding systems have been analyzed, identifying user interaction and fail-safe design gaps [5]. The Chicken Nanny's focus on standalone operation addresses those concerns by providing physical configuration through a keypad and EEPROM persistence.

Environmental control remains another active area. An IoT-based environmental controller for poultry houses using a robot platform has been introduced [6], while LoRa-based smart water meters have been explored [7]. Both highlight sensor integration and wireless feedback, but still rely on external networks for full functionality. In contrast, our system achieves temperature regulation with onboard thermocouple feedback, maintaining reliability even in offline scenarios.

A particularly relevant agricultural system is a machine vision-based salted egg cleaning and grading system using a Raspberry Pi and OpenCV [8][9]. While the use case differs from poultry feeding and heating, integrating embedded hardware with real-time control and classification showcases the effectiveness of standalone automation solutions in agriculture. This aligns closely with our project's aim to deliver reliable, affordable, and modular automation for small-scale farming.

Non-agricultural projects also offer insight into embedded systems and modular automation [10]. Temperature changes for ship sensors have been simulated using a microcontroller-based rig [11], showing comparable thermal feedback handling. Likewise, an Arduino ultrasonic sensor system for fluid concentration measurement is presented [12][13], demonstrating microcontroller versatility in diverse settings.

Mid-air augmented reality (AR) systems [14] have been explored for robot communication. While outside the agricultural scope, the project's emphasis on modular integration and novel user interfaces inspired aspects of the Chicken Nanny's standalone configurability.

Together, these works frame various challenges in automating poultry care, from environmental monitoring to actuator control and user interface design. The Chicken Nanny builds on these insights by integrating feeding, heating, and input in a single system tailored to small-scale poultry environments without external computing or network reliance.

III. IMPLEMENTATION

A. Software Approach and Implementation

The software for The Chicken Nanny was implemented using Arduino C++, with modular functions designed to manage real-time feeding schedules, ambient temperature regulation, and interactive configuration via a keypad. The control logic is hosted on a REXQualis Uno R3 (ATmega328P) and structured around cooperative multitasking using non-blocking loops.

At startup, the program initializes all peripherals, including the DS3231 Real-Time Clock (RTC) for precise timekeeping via I2C, the MAX6675 thermocouple with a K-type probe for ambient temperature sensing over SPI, and the SG90 servo motor to control the feed gate. It also sets up a 5V relay to switch a 120V 200W radiant heater, a 16x2 I2C Liquid Crystal Display (LCD) to show system status and user prompts, and a 4x4 matrix keypad, which is read using a keypad scanning library, for real-time configuration input.

The software workflow can be described as follows:

(1) The system relies on EEPROM-based configuration storage. During the initial startup or when initiated by the user, it enters a setup mode that allows configuration of the feeding schedule in hours and minutes, the number of chickens to calculate the feed quantity, and the heater activation temperature. These settings are entered through the 4×4 matrix keypad and shown in real time on the LCD. Once confirmed, the configuration data is saved to EEPROM so it is retained across power cycles.

(2) The feeding control logic continuously checks the current time from the real-time clock DS3231 and compares it with the stored feeding schedule. When the time matches the configured feeding

period, the servo motor is driven to rotate the gate into the open position. A non-blocking timing mechanism tracks how long feeding has been active, after which the servo returns the gate to the closed position. Special conditions, such as avoiding repeated activations within the same minute, are handled through the use of state flags and time comparisons.

(3) The heating control logic obtains ambient temperature measurements from the thermocouple with a resolution of 0.25 °C. These readings are continuously evaluated against two defined thresholds: when the temperature falls below the activation point, the relay is engaged to switch the heater on, and when the temperature rises above the reset point, the relay is disengaged to switch the heater off. This hysteresis-based method prevents rapid relay cycling near the threshold and improves overall system stability.

(4) The LCD display is refreshed every second to show the current time, the current ambient temperature, and the status of the heater.

(5) The keypad menu system allows the user to enter an edit mode by pressing a designated key (*). Within this mode, the directional keys (2, 4, 6, and 8) are used to move between adjustable parameters, numeric keys (0–9) are used to enter new values, and the confirm key (#) saves the input and proceeds to the next setting. If no key input is detected for a predefined period, the system automatically exits the menu and resumes normal operation. Key debouncing and input buffering are managed by the keypad library to ensure reliable and smooth user interaction.

(6) The system loop is structured around 150-millisecond delay intervals to manage timing, enabling simultaneous monitoring of time, temperature, user input, and actuator status without interruption. Within each cycle, the loop performs essential tasks such as regulating the feeder and heater, processing keypad input, refreshing the display, and loading or prompting configuration data. Each task is governed by conditional checks (such as flags or timed intervals) to ensure the system remains responsive.

This structure ensures that the Chicken Nanny operates reliably and can be easily adapted or expanded to support additional automation features such as door control, data logging, or wireless alerts without requiring significant restructuring.

B. Hardware Approach, Equipment, and Programming

The hardware for the Chicken Nanny was designed to deliver robust, flexible control over two primary coop functions: automated feeding and temperature regulation. The system is built around a REXQualis UNO R3 microcontroller board using the ATmega328P chip. All components are housed inside a 12” x 12” plastic junction box for protection from dust, debris, and accidental contact, with wiring routed cleanly using solid-core jumper wires to maintain secure connections. The hardware design is shown in Fig. 1 below.

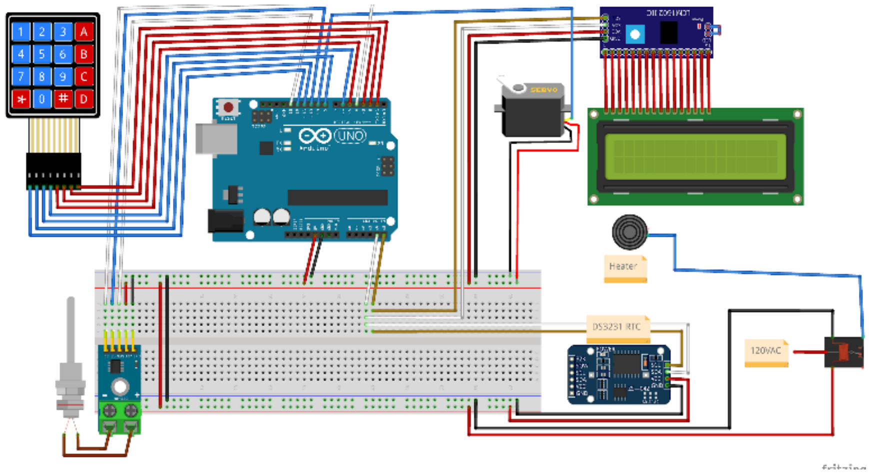

Fig. 1. System Interface

The Arduino is powered by a 9V regulated DC adapter capable of supplying sufficient current for all connected peripherals, including the servo motor and relay module.

A DS3231 real-time clock module, connected via I2C, provides highly accurate timekeeping for scheduled feeding operations. The system checks the current time once per minute and triggers the feeding mechanism when it matches the user-defined feeding schedule.

To automate feeding, a SG90 servo motor is used to rotate a 3D-printed gate [15], shown in Fig. 2, covering the feed chute. When feeding time arrives, the servo rotates to an open position for a set number of seconds—determined by the number of chickens—before returning to the closed position. This ensures a consistent and predictable amount of feed is dispensed. The gate's motion was calibrated and tested for speed and reliability during physical prototyping.

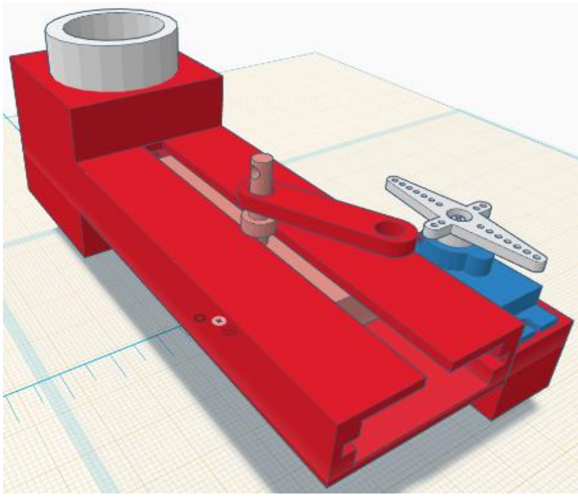

Fig. 2. 3D-Modeled Servo-Controlled Feed Gate

Ambient temperature is measured using a MAX6675 thermocouple module paired with a K-type thermocouple probe. This module uses SPI communication to provide high-resolution temperature readings every few seconds. When the temperature drops below the user-defined heating threshold, the system energizes a 5V relay module to switch power to a 120V, 200-watt flat panel radiant heater. The relay deactivates the heater once the temperature rises above a separate reset threshold. This hysteresis model prevents constant toggling near the boundary temperature.

The high-voltage components, including the relay and heater, are electrically isolated from the low-voltage control circuitry to protect the Arduino and ensure user safety. The relay is used to switch only the hot conductor of the AC supply, while the neutral and ground connections are hardwired to reduce electrical interference. The relay system configuration is shown in Fig. 3.

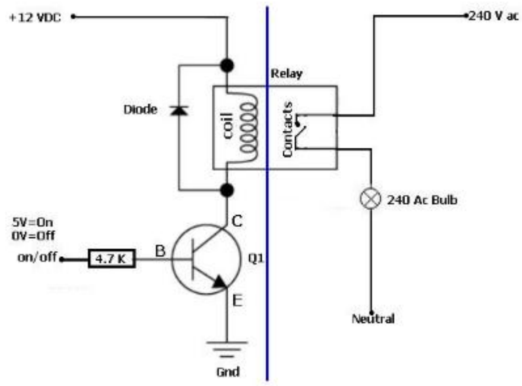

Fig. 3. Circuit Diagram of Relay System

System status is displayed in real-time using a 16x2 I2C LCD, which shows the current time and ambient temperature. A 4x4 membrane matrix keypad lets the user input and update configuration values directly from the device. The keypad uses a scanning algorithm and is read using an interrupt-safe polling routine, with visual feedback provided via the LCD.

At startup, the Arduino retrieves configuration settings such as feeding time, feeding duration, and temperature thresholds from EEPROM. If no stored values are detected or if the user initiates edit mode through a keypad input, the system enters a setup routine. The user can step through each parameter, enter new values, and confirm the settings using the keypad, with all updates saved to EEPROM to ensure they persist across power cycles.

The Arduino software is organized into a modular structure in which one section manages the initialization of all hardware components, while the main execution cycle repeatedly performs non-blocking tasks such as time monitoring, temperature sensing, actuator control, and user input processing. All time-dependent operations rely on elapsed-time tracking rather than blocking delays, ensuring the system remains responsive while waiting for timers or user interactions.

Overall, this hardware setup prioritizes low cost, ease of assembly, and flexibility, making it ideal for small-scale poultry operations or hobbyist environments. The modular design also provides a foundation for future expansions, such as automated coop doors, network integration, or solar power support.

IV. EVALUATION

The Chicken Nanny system was tested through several iterative trials to validate its feeding precision, heating responsiveness, and interface reliability under conditions approximating real-world usage. The design process also involved evaluating alternative mechanisms and making informed compromises based on system complexity, material constraints, and project scope.

Initial design exploration considered a DC motor-driven dispensing wheel, modeled after mechanisms in devices like gumball machines, due to its potential for high volumetric accuracy. However, the added mechanical complexity and control requirements made the approach impractical for a rapid prototype. The final design instead uses a sliding gate operated by a servo motor, selected for its simplicity and predictable motion between fixed open and closed positions.

A prototype was constructed using an inverted 2-liter soda bottle fitted with a cardboard gate to validate the gate design and feed flow behavior. Five test runs using uncooked rice as a stand-in for feed demonstrated that the flow rate remained relatively constant regardless of remaining volume—an important consideration, as real-world feed bins naturally decrease over time.

This concept was translated into a CAD model using TinkerCAD and then 3D-printed. During servo integration, the open and closed angles had to be calibrated due to initial misalignment, which caused incomplete gate movement. After the adjustment, the system consistently reached both endpoints. Additional iterations were done to tune servo speed and gate timing, as the feed was initially dispensed too quickly, leading to overfeeding.

Design considerations also included jam prevention. The housing was constructed with 0.5 to 1 mm of clearance around the gate, and a lateral ejection path was added to allow partially trapped feed to escape during the closing motion. The final system was tested using actual chicken feed—a blend of pellets and crumble—and consistently delivered feed without clogging or performance degradation.

The ambient temperature sensing and heater control subsystem was validated by immersing the thermocouple in an ice bath to confirm its low-end accuracy, followed by observing system response under room temperature conditions. The thermocouple's readings matched expected values within a tolerable range, and the heater was verified to activate below the configured threshold using an inline LED to indicate relay status.

The control algorithm includes a 5°F hysteresis window between activation and deactivation thresholds, preventing relay chatter or rapid switching. This behavior was consistent during bench testing, and the relay reliably switched the 120V heater on and off as intended.

The DS3231 RTC module performed reliably throughout the testing period. Scheduled feeding occurred at the correct minute without noticeable drift, and all time-based triggers were executed predictably. System configuration was retained across power cycles using EEPROM.

The keypad interface allowed users to configure feed timing, duration, and temperature thresholds directly from the device. Input accuracy and responsiveness were confirmed over repeated use. One issue during early testing was flickering on the I2C LCD screen, traced to intermittent connections from low-quality jumper wires. This was resolved by switching to solid-core wiring and reseating all I2C connections.

While the Chicken Nanny was not benchmarked against commercial poultry automation systems, early research revealed a lack of low-cost, open-source solutions with both feeding and heating integration. Most small-scale automation efforts focus on either lighting or coop door automation. The system developed in this project provides a unique combination of affordability, configurability, and mechanical robustness, with no external computing required for normal operation. The Chicken Nanny system is shown in Fig. 4.

Fig. 4. The Chicken Nanny System

V. CONCLUSION AND FUTURE WORK

This research focused on automating two critical functions in small-scale poultry farming, timed feeding and temperature-regulated heating. Manual management of these tasks is often labor-intensive and inconsistent, especially for hobbyists and rural farmers who lack access to expensive commercial automation systems. The proposed solution, the Chicken Nanny, utilizes an Arduino Uno based platform to automate feeding and heating through a servo operated gate, thermocouple-based temperature sensing, and relay-activated heating, all contained within a compact and protected control enclosure.

Future work will focus on expanding user configurability through the keypad interface. Planned enhancements include adding and removing feeding times, interactively setting feeding durations, and assigning dedicated keypad buttons for specific actions, such as adding a feeding time or adjusting duration. The existing software architecture already provides modular support for managing user input and configuration workflows. Additional improvements may include clearer user feedback during active feeding or heating cycles through LCD messages or LED indicators. In the longer term, the team envisions developing a companion mobile application to enable remote monitoring of coop conditions and system status, providing added value to end users.

REFERENCES

[1]. D. Karastoyanov, E. Blagoeva, and D. Yarkov, "Innovations in Robotic Animal Husbandry," *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, Crete, Greece, pp. 300-303, doi: 10.1109/CSCC55931.2022.00058, (2022).

[2]. K. P. Yang, P. McDowell, P. Dolan, C. Otts, G. Chenevert, and C.Tunstall, “SelfieBot: A Robot to Detect, Photograph, and Tweet Smiles,” *International Journal of Engineering Research & Technology (IJERT)*, ISSN 2278-0181, vol. 8, issue 10, pp. 387–390, (2019).

[3]. J. Bea and J. Cruz, “Chicken Farm Monitoring System Using Sensors and Arduino Microcontroller,” *Proceedings of the 9th International Conference on Information Systems and Technologies (ICIST’19)*, Article No. 28, pp. 1-4, doi: 10.1145/3361570.3361607, (2019).

[4]. P. N. Vrishanka, P. Prabhakar, D. Shet, and K. Rupali, "Automated Pet Feeder using IoT," *2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, Tumkur, Karnataka, India, pp. 1-5, doi: 10.1109/ICMNWC52512.2021.9688391, (2021).

[5]. E. Blagoeva, B. Karkov and N. Stoimenov, "Review and Analysis of Robotized Feeding Systems," *2021 International Conference Automatics and Informatics (ICAI)*, Varna, Bulgaria, pp. 341-344, doi: 10.1109/ICAI52893.2021.9639549, (2021).

[6]. H. Sun, T. D. Palaoag, and Q. Quan, “Design of Automatic Monitoring and Control System for Livestock and Poultry house Environment Based on Internet of Things robot,” *Proceedings of the 2022 4th Asia Pacific Information Technology Conference (APIT’22)*, pp. 224-230, Thailand, doi: 10.1145/3512353.3512386, (2022).

[7]. Z. Lei, Y. Xie, and X. Jiang, "Design and Implementation of Smart Iot Water Meter Based on LoRa," *Proceedings of the 2024 8th International Conference on Electronic Information Technology and Computer Engineering (EITCE ‘24)*, Haikou, Hainan, China, pp. 1447-1451, doi: 10.1145/3711129.3711371, (2024).

[8]. K. P. Yang, N. Moran, I. Bendana, S. Champagne, and T. Becker, "LOPEZ: A Bilingual Robotic Car*," International Journal of Research in Advent Technology (IJRAT)*, E-ISSN 2321-9637, vol. 4, issue 12, pp. 51-55, (2016).
[9]. L. Bengua, V. Guzman; D. Macunat; E. Villaverde; A. Mahusay, and R. Maaliw, "Salted Egg Cleaning and Grading System Using Machine Vision," *2022 IEEE World AI IoT Congress (AIIoT)*, Seattle, WA, USA, pp. 489-493, doi: 10.1109/AIIoT54504.2022.9817366, (2022).
[10]. K.P. Yang, G. Kiepper, B. Henry, and R. Hunter, "Modular Architecture for IoT Home Automation and Security Surveillance," *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, ISSN 2458-9403, vol. 5, issue 11, pp. 8978–8982, (2018).
[11]. L. Cao, K. Zhang, Y. Hu, Y. Luo, W. Su, R. Tang, and G. Zhou, "Design of Temperature Simulation Device for Military Ship Sensors," *Proceedings of the 2023 5th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI '23)*, Nanchang, China, pp. 621-625, doi: 10.1145/3653081.365318, (2023).
[12]. K.P. Yang, P. McDowell, P. Devkota, S. Pradhan, R. Bhandari, and Z. Madewell, "Detecting Gas Leaks: A Case Study in IoT Technologies," *European Journal of Engineering and Technology Research (EJ-ENG)*, ISSN 2736-576X, vol. 6, issue 7, pp. 103–106, doi:10.24018/ejeng.2021.6.7.2663, (2021).
[13]. R. Xu, Z. Liu, S. Zhang, T. Ke, and T. Guo "The Measurement of Liquid Concentration Based on Arduino UNO Platform Using Ultrasonic Wave Sensor," *Proceedings of the 2023 6th International Conference on Signal Processing and Machine Learning (SPML '23)*, pp. 253-259, doi: 10.1145/3614008.3614047, Tianjin, China, (2023).
[14]. A. Lozada, U. Tijani, V. Keth, H. Wang and Z. Han, "Anywhere Projected AR for Robot Communication: A Mid-Air Fog Screen-Robot System," *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Melbourne, Australia, pp. 520-528, doi: 10.1109/HRI61500.2025.10974070, (2025).
[15]. K. P. Yang, P. McDowell, R. Demourelle, T. Parker, and E. Langstonirst, "3D Printing: A Custom-Built 3D Printer with Wireless Connectivity," *SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE)*, ISSN 2348 – 8387, vol. 7, issue 10, pp. 1-5, doi: 10.14445/23488387/IJCSE-V7I10P101, (2020).