EasyChair Preprint

№ 16001

IoT-Driven Smart Management in Broiler Farming: Simulation of Remote Sensing and Control Systems

Sandra Isabella Coello Suárez, V. Sanchez Padilla,
Ronald Ponguillo-Intriago and Albert Espinal

September 7, 2025

IoT-Driven Smart Management in Broiler Farming: Simulation of Remote Sensing and Control Systems

Sandra Coello Suarez[1], V. Sanchez Padilla[2,3], Ronald Ponguillo-Intriago[4], and Albert Espinal[1]

[1]Telematics Engineering Dept., Escuela Superior Politécnica del Litoral, Guayaquil, EC090112, Ecuador
[2]Dept. of Engineering Education, Virginia Tech, Blacksburg, VA 24060, USA
[3]Faculty of Engineering, Universidad ECOTEC, Samborondón, EC092302, Ecuador
[4]Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil, Guayaquil, 090313, Ecuador
{saiscoel, aespinal}@espol.edu.ec, vsanchez@vt.edu, rponguil.point@gmail.com

Abstract—**Parameter monitoring and control systems are crucial in the industry as they enable automation processes that improve productivity and resource optimization. These improvements also help to manage environmental factors and the complex interactions between multiple inputs and outputs required for production management. This paper proposes an automation system for broiler management based on a simulation scenario that involves sensor networks and embedded systems. The aim is to create a transmission network for monitoring and controlling broiler temperature and feeding using the Internet of Things (IoT), complemented by a dashboard and a cloud-based service database to track improvements in broiler management. We look forward this work will serve as a guide for stakeholders and entrepreneurs in the animal production industry, fostering sustainable development through simple and cost-effective automation solutions. The goal is for them to scale and integrate these recommendations into their existing operations, leading to more efficient decision-making at the management level.**

Index Terms—**broiler management, monitoring system, poultry productivity, quality food.**

I. Introduction

Many farmers responsible for broiler breeding use manual or informal methods without automation due to a lack of proper training in technology or logistics [1]–[3]. Since broilers are a huge source of food for people, both for their meat and derivatives, they also represent a valuable income for farmers. Important parameters in broiler breeding include temperature and food supplementation by time intervals, in which control can be challenging based on available resources and assets [4], [5]. An appropriate ambient temperature should follow the recommended range or thresholds, or broiler breeding will be directly or indirectly affected by the wrong setting of these critical parameters, resulting in animal stress [6]–[8]. Farmers should also consider levels of food supply, for example, if it is low, it leads to a low weight group, with unsuitable productivity outcomes [5]. Excessive eating and an excess of nutrients provided to the broiler also lead to inappropriate fattening, which could have negative consequences for broiler management [9].

A way to address the problem described is to set up a monitoring system to visualize and control environmental temperature and food distribution with alerts [10]–[12]. Current technology allows more options and ease of connection to the Internet to support process automation, which enhances production. The Internet of Things (IoT) includes network connectivity concepts to improve transmissions and control in home and industrial systems through protocols and technologies to enhance productivity, increase efficiency, reduce cost production and meet business demands [13], [14]. Thus, we propose a wireless sensor network simulation to monitor and control broiler production through a network system that triggers warnings based on changes in ambient temperature thresholds and food supply.

The first stage of this work consists of sensors that transmit data on temperature and food load values for food storage and monitor the breeding process to optimize resources. Sensors control the threshold temperature based on a reading process in which the system verifies the temperature by comparing the values for a subsequent activation of the actuator, such as a fan or lamp, to stabilize the proposed temperature. The second stage enables us to know if food is required to supply the broiler based on the results of a load sensor that measures its weight. An Arduino transmits data to a Raspberry Pi through serial communication to save it and present it on a dashboard to allow users to access the data via a website for monitoring, remote control processing, and retrieval for statistical analysis. The temperature limits may change depending on certain factors of broiler management.

Our simulation proposal applies to tropical coastal areas at sea level, replicable to other environments based on specifications asked by broiler farmers. The purpose is to offer an IoT-based solution for broiler management through simulation with ICT tools to optimize automation processes for food distribution and temperature control to improve productivity by focusing on good practices, ensuring access to quality food for the population and responsible rural production, outcomes that align with the [1]United Nations Sustainable Development Goals 2 (Rural Development, Food Security and Nutrition) and Goals 12 (Sustainable Consumption and Production).

The organization of this work has the following structure: Section II presents a short related work on systems that approach ICT for broiler production. Section III focuses on

[1] https://www.undp.org/sustainable-development-goals

the methods used for the simulation of the proposed network system, describing the materials used and the integration of the components. The results of the system tests and simulation scenarios are presented in Section IV. Finally, Section V presents the conclusions.

II. Related Work

Studies focused on technical-level broiler management present different features and approaches. For instance, Elham et al. [15] monitor the environment of poultry farms using IoT and Blockchain. The methodology consisted of acquiring information through IoT-based temperature sensors for data processing in JSON format using IOTA, an integrated technology between IoT and blockchain. This method allows computers on the IOTA network to convey data and immutable values to each other. Similarly, Watthanawisuth et al. [16] introduced ZigBee technology for a wireless interconnection of a sensor network with the advantage of operating at low energy consumption. In this case, the system monitors the air temperature microclimate with a high degree of coverage through a cluster tree topology to ensure the necessary scope using a few sensor nodes. The microcontroller interacts with duly protected temperature sensors by a cylindrical casing to avoid errors during the variables' or interference measurements. A photoelectric cell energizes the sensors through an energy storage circuit to allow the sensors to work even in the evenings, and a receptor processes the data in determined periods depending on the power required by each node.

Choukidar and N. Dawande [17] employed sensors for level measurements of food, gas, water, and temperature to maintain environmental conditions within the farm. They connected the sensors to a Raspberry Pi for data monitoring and GPRS-based control to allow the farm administrators to consult the data gathered by the sensors through a web application. Following automation processes, they kept the parameters in the appropriate ranges by combining logical and physical elements. Their findings include an incidence reduction of environmental conditions that affect the weight and health of the poultry, leveraging an opportunity to save costs. Thus, the cases presented in this section approach to applying microcontrollers and microprocessors in interaction with sensors to convey information through wireless technologies to enhance farming processes.

III. Methods

We implemented a single-node configuration using a temperature sensor without communication between more sensors because of the potential of the adaptability of radio modules for data conveying to establish transmission with microcontrollers and microprocessors. Further, photovoltaic cells supply energy for sensors to measure temperature and food loading and show information through end-user interfaces from the web application, such as Ubidots, allowing an interaction with our devices in a programmatic way through a friendly user interface with cloud-based services for IoT [18]–[20]. The importance of these parameters lies in their direct influence on the breeding in which the information transmission is through a Wi-Fi module to attain lower budgets. A similarity of our work with related literature is the star topology-based deployment to achieve communication with a central node and data conveying. We considered this topology convenient for our network system because of its advantage of operating at low power consumption [21]. We relied on the cloud for later dashboard visualization, as depicted in Fig. 1.

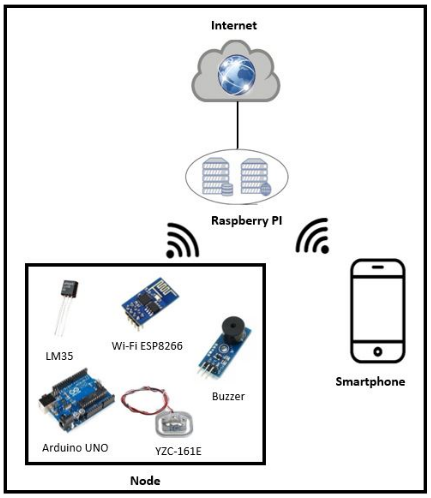


Fig. 1: System topology.

A. Materials

We deemed the simulation scenario on the materials described because they are convenient for the conditions presented, representing low-cost alternatives for the purpose we followed. However, other approaches with different requirements can use similar software and hardware. The materials are listed as follows:

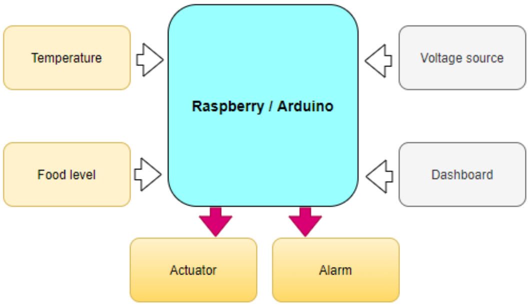


Fig. 2: Block diagram.

- [2]LM35: Sensor that has a calibrated precision of 1 linear centigrade degree. It measures the temperature of the environment and returns a value in volts. The ratio of 1°C equals 10mV [22].
- [3]YZC-161E: Sensor to measure either the load or weight of an object. It has a nominal load of 5 kg with an

[2] https://www.ti.com/document-viewer/LM35/datasheet

[3] https://datasheethub.com/yzc-131-10kg-range-weighing-sensor-load-cell-sensor/

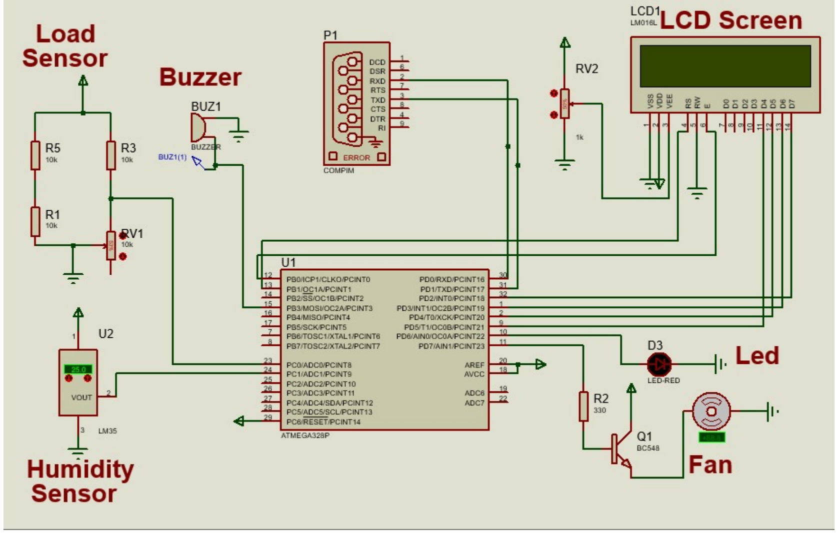


Fig. 3: Proteus sensor node diagram.

operating temperature range of -21 ∼ 40°C. It is recommendable to work with an excitation voltage of 5 Vdc and a weight of 80 g.

- Arduino UNO: Microcontroller for running and controlling programmed activities via sensor interaction [23]. It comprises an ATmega328 chip, 5V operating voltage, direct current per I/O Pin of 40mA, 32 KB Flash memory, and 2 KB SRAM.
- ESP8266: A Wi-Fi module for wireless communication between modules through telemetry protocols [24], [25]. It uses a Tensilica L106 32-bit CPU, 3V to 3.6V operating voltage, and 80 mA. It supports IPv4 and TCP/UDP/HTTP/FTP protocols.
- Raspberry PI 3 model B+: Electronic development board for information processing and analysis. It comprises a CPU + GPU with Broadcom BCM2837B0 card, 64 bits through 1.4GHz, with 1GB LPDDR2 SDRAM RAM. Its communication circuitry supports 2.4GHz and 5GHz bands, with IEEE 802.11 and Bluetooth protocols [26].
- Buzzer: A module for alerting through sound with a 5Vdc passive buzzer, three pins, VCC +, GND -, and I/O signal [27]. It requires a frequency of at least 1.5 kHz. Its advantage lies in its compact size.

B. Modular integration

Fig. 2 shows a block diagram with the interaction of the functionalities of the components. The temperature and food level sensors convey the data obtained to the Raspberry/Arduino module for information processing. Based on the configured values, an alarm triggers to indicate the action to take, i.e., if the temperature values are very high, a mechanism activates to regulate the temperature. Likewise, the system will fill the reservoirs to supply food to the broiler if the food level in the dispenser is low. The data gathered are saved and displayed in a web application through a dashboard.

The sensor node integrated the temperature sensor, load sensor, buzzer module, Wi-Fi interface, and Arduino board. The LM35 sensor provides an output voltage that is proportional to the temperature being measured [22], allowing us to measure the temperature integrating the Arduino, Raspberry PI, and Nodemcu IoT platform [28]. The power control works through the YZC-161E Strain Gauge, setting it at 50 kg, while the strain gauge enables a transducer to support compression, tension or bending loads to convert force into an electrical signal.

The sensors were located at the bottom of the reservoirs to measure the food storage level, in which the reservoirs store the food with no more than 50 kg of weight, alerting the user if they weight less than 10 kg to indicate the food storage restocking. The LM35 and YZC-161E sensors communicated with the Arduino UNO board to assess if the temperature and food level parameters were correct. Otherwise, it sends the signal to the buzzer module to issue an audible alert for a recalibration set by the administrator. The Raspberry Pi received the data through the ESP8266 Wi-Fi module to make possible the visualization of the web application to depict the variable status, i.e., the temperature and the food level to supply the broiler.

The integration of the sensor as a node allowed measuring of the temperature and food distribution controlled by the Arduino UNO, which saved the information for transmission to the Gateway node. It is possible to verify the functionality of the Gateway node at the web and database service level implemented in the Raspberry Pi [29]–[31]. In our case, we chose to work on Raspbian Linux to setup the graphical interface for the IoT-based embedded system [32].

We used Proteus Design Suite software because it allowed electronic design automation. Fig. 3 shows the schematic diagram of the temperature and food level monitoring system. The LM35 measured the temperature through a variation-based comparison, sending an analog signal convert it to a digital

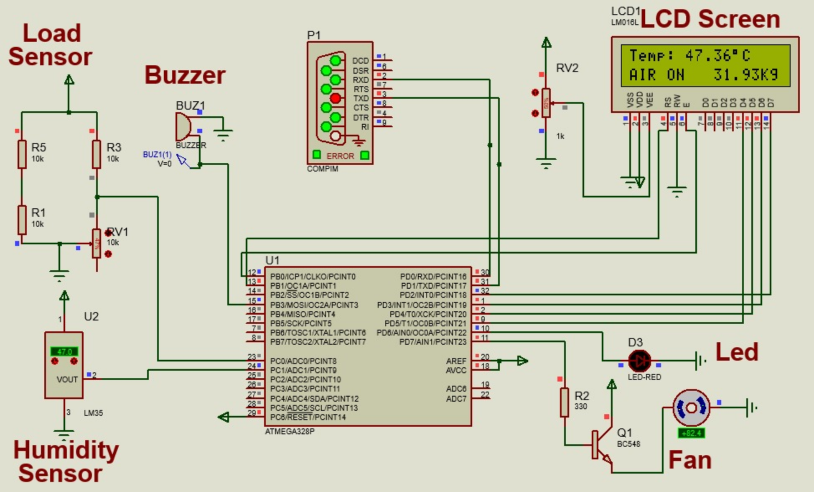


Fig. 4: First scenario simulated

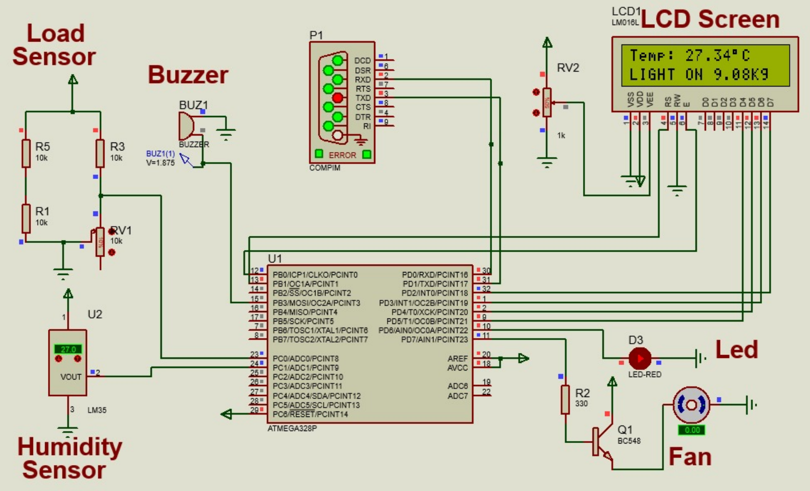


Fig. 5: Second scenario simulated

value. We have used an array of resistors to simulate the strain gauges. In one of them, we considered a potentiometer to vary the loading through an analog value to set the maximum and minimum values to obtain the results in kilograms.

IV. Results

A. System testing

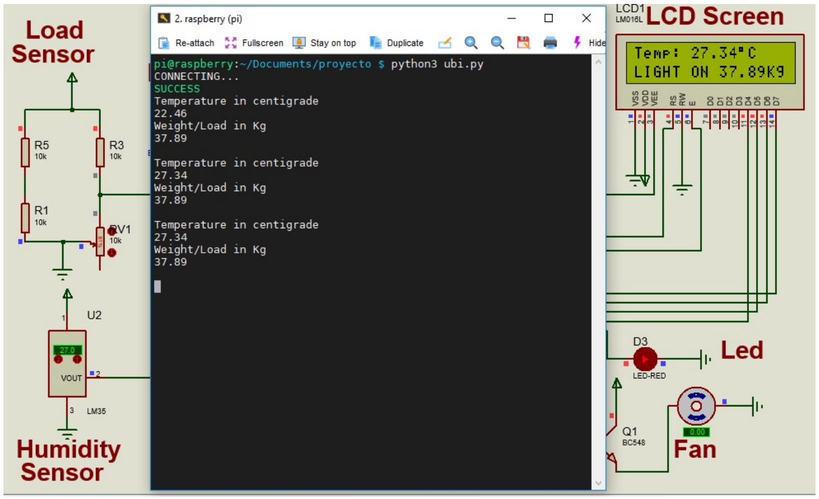
The network system has visual and sound indications to optimize broiler management. We established an ideal temperature of 39°C. If the temperature got lower, the function turned on a light bulb. In the diagram, the LED represents this function, so if the temperature was higher than the ideal, a fan represented by a motor turned on. For the food loading in the dispenser, we established a minimum value of 10 Kg. If the sensor reads less than this value, the system emits a sound alert through a buzzer for a determined time.

The Proteus COMPIM simulated the sensor node connectivity with the Raspberry Pi through serial communication. We can remark that the LCD allowed visual indication of the data gathered by the sensors. In addition, Ubidots performed as a dashboard by device creation, and a token conveyed the data to this module. We set two variables called Load and Temperature, each with an ID used in the coding to know which variable transmitted the temperature and loading data.

B. Scenarios simulated

The first scenario consisted of an environment at 47.36°C. Since this temperature was higher than 39°C (ideal room temperature), the LED remained off, and the motor (fan) turned on to decrease the room temperature. Fig. 4 depicts the fan speed in revolutions per minute (RPM) from the Proteus simulation. The buzzer did not turn on since the food amount level was above 10 kg. Fig. 5 presents a second scenario with a temperature of 27.34°C (less than 39°C). The LED did light up with the motor at 0 RPM. The buzzer remained off since it did not meet the conditions for turning it on.

We reduced the potentiometer value for shifting the food loading level to 9.08 kg (less than 10 kg). It caused the buzzer

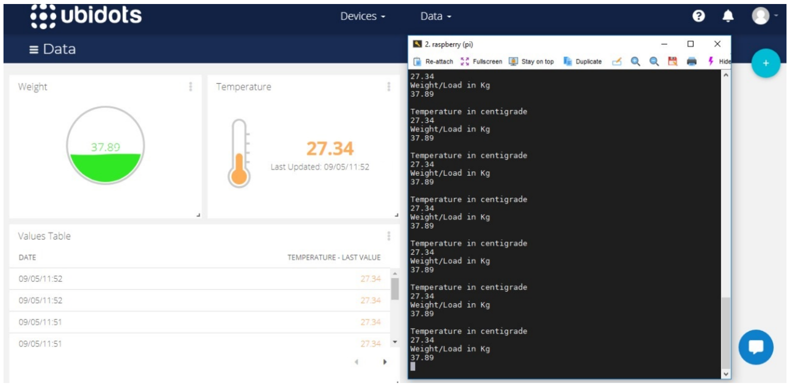


```
pi@raspberry:~/Documents/proyecto $ python3 ubi.py
CONNECTING...
SUCCESS
Temperature in centigrade
22.46
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89
```

Fig. 6: Data reception



DATE	TEMPERATURE - LAST VALUE
09/05/11:52	27.34
09/05/11:52	27.34
09/05/11:51	27.34
09/05/11:51	27.34

```
27.34
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89

Temperature in centigrade
27.34
Weight/Load in Kg
37.89
```

Fig. 7: Dashboard

to emit a 2-second interval sound, which remained in alert mode until loading the food dispenser again. We configured a Python script that received the sensor data through a serial connection with Proteus on a virtual machine with Raspbian. The script transmitted the data to Ubidots through the API client for a dashboard visualization. Fig. 6 shows a transition between states, such as connecting and success, representing the connection between Proteus and Ubidots. The temperature and loading values were in kilograms and displayed through the console. The system provided real-time updates to check the food loading and temperature through the dashboard, as shown in Fig. 7.

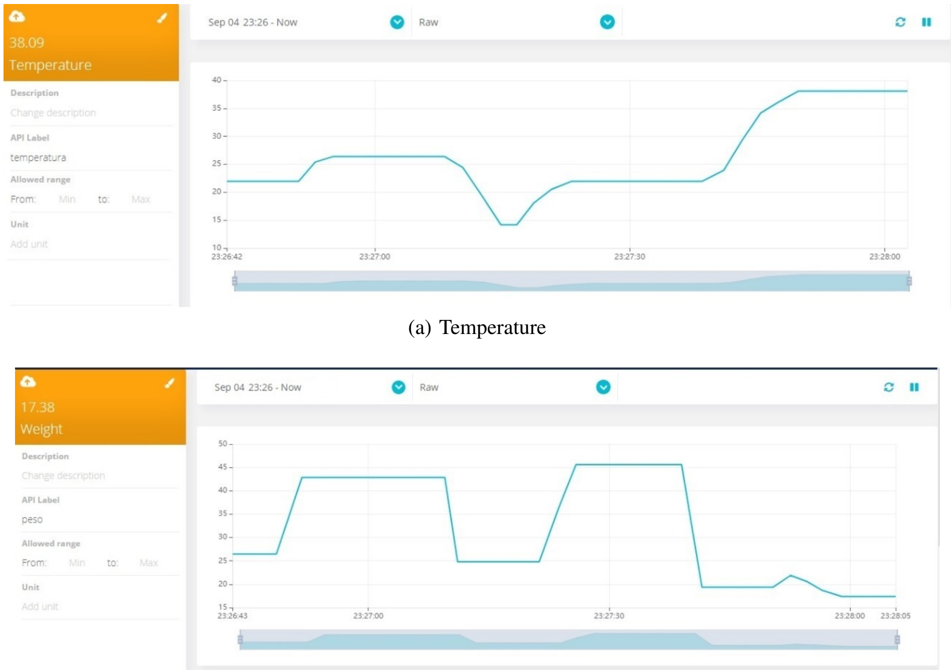
C. Analysis

This work simulated a broiler farm environment based on experiments. Therefore, the data gathered can resemble contexts where the temperature and load sensors undergo similar conditions. Fig. 8 represents the temperature obtained in two days (8a) and the values of food load in the same period (8b). It is possible to export the results gathered from Ubidots during the simulation into a *.csv* format and perform data filtering for specific intervals. A script allows manipulating the data collected in the *.csv* file with 1103 values on different schedules. The sum of data from a day was gathered and divided by the amount of data to obtain the mean for two consecutive days. On the first day, the mean was approximately 38.61°C and 40.63°C on the second day, as depicted in Fig. 9. *Appendix A* lists the Raspberry coding to integrate the results collected from Ubidots with the API for connection testing and to print the temperature and food load results.

On both days, we expected a mean of 39°C to represent adequate conditions for broiler management. However, since the data taken to obtain the displayed mean comes from the direct manipulation of people, we do not recommend a comparison with an average taken from field implementations. We verify that our work meets the proposed objectives, such as issuing visual and sound alerts and capturing changes in the variables of primary interest. Thus, we contributed to resource optimization and ensured conditions for recommendable broiler management.

V. Conclusion

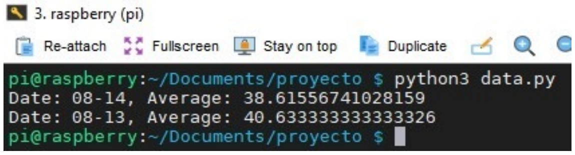
The simulation yielded accurate results for the proposed objectives, as the temperature and load variables responded effectively to the external factors detected by the system. However, this proposal must also adhere to the regulations and



(a) Temperature

(b) Food load

Fig. 8: Screenshot of the results



```
pi@raspberry:~/Documents/proyecto $ python3 data.py
Date: 08-14, Average: 38.61556741028159
Date: 08-13, Average: 40.633333333333326
pi@raspberry:~/Documents/proyecto $
```

Fig. 9: Two-day monitoring mean (average)

policies of the implementation site. Depending on the context, the integration of robust sensors can ensure reliable data acquisition. This does not necessarily lead to higher equipment expenditures, but may require investment in training for managers and staff. It is important for the readers to consider that higher-capacity electronic development boards (e.g., FPGAs) can enhance implementation dynamics, particularly when it comes to sensor integration. These boards align with wireless communication concepts, supporting components that transmit through different protocols and channel frequencies.

In addition, a reliable circuit power supply must always be in place to handle power failure events, incorporating backups such as a battery bank or uninterrupted power sources. Actuator and alert element tests should include a backup system, such as SMS warning logs, to verify the information displayed on the dashboard, either during a failure or inaccessibility. For future work, we recommend integrating other open-source technologies to address budget concerns, especially if a deployment based on our guidelines needs to expand its scope and follow specific rules.

References

[1] M. S. Istiak and A. Khaliduzzaman, “Poultry and egg production: An overview,” *Informatics in Poultry Production: A Technical Guidebook for Egg and Poultry Education, Research and Industry*, pp. 3–12, 2022.

[2] S. Bhanja, V. Sahu, *et al.*, “Automation and smart poultry farm management: A review,” *Indian Journal of Poultry Science*, vol. 56, no. 2, pp. 89–99, 2021.

[3] E. Sonaiya and S. Swan, “Small-scale poultry production, technical guide,” 2004.

[4] J. Astill, R. A. Dara, E. D. Fraser, B. Roberts, and S. Sharif, “Smart poultry management: Smart sensors, big data, and the internet of things,” *Computers and Electronics in Agriculture*, vol. 170, p. 105291, 2020.

[5] D. Cassuce, I. Tinôco, F. Baêta, S. Zolnier, P. Cecon, and M. d. F. Vieira, “Thermal comfort temperature update for broiler chickens up to 21 days of age,” *Engenharia Agrícola*, vol. 33, pp. 28–36, 02 2013.

[6] V. Šranková, J. Lendelová, M. Žitňák, I. Karandušovská, T. Szabóová, Š. Mihina, M. Balková, and Š. Pogran, “Floor temperature as a risk factor for the quality of the environment in the chickens,” *Agronomy Research*, vol. 17, no. 6, pp. 2428–2437, 2019.

[7] A. Nawab, F. Ibtisham, G. Li, B. Kieser, J. Wu, W. Liu, Y. Zhao, Y. Nawab, K. Li, M. Xiao, *et al.*, “Heat stress in poultry production: Mitigation strategies to overcome the future challenges facing the global poultry industry,” *Journal of thermal biology*, vol. 78, pp. 131–139, 2018.

[8] D. Moura, R. Vercellino, J. Santos, and M. Vale, “Heat stress impact on weight gain in broiler chickens: a meta-analytical study of environmental factor that impact production losses,” in *ASABE 1st Climate Change Symposium: Adaptation and Mitigation*, 05 2015.

[9] M. Kucheruk, S. Midyk, D. Zasekin, V. Ushkalov, and Kepple, “Comparison of fatty acid content of organic and traditionally grown broiler chickens,” *Food Science and Technology*, vol. 13, no. 4, pp. 51–57, 2019.

[10] A. Shanmugapriya, A. Sangeethadevi, and A. Kalaivani, “Poultry farm surveillance system utilizing IoT and wireless sensor network,” in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, pp. 1061–1066, 2022.

[11] F. J. Adha, R. Ramli, M. H. Alkawaz, M. G. M. Johar, and A. I. Hajamydeen, “Assessment of conceptual framework for monitoring poultry farm’s temperature and humidity,” in *2021 IEEE 11th International*

Conference on System Engineering and Technology (ICSET), pp. 40–45, 2021.

[12] E. Edwan, M. A. Qassem, S. A. Al-Roos, M. Elnaggar, G. Ahmed, A. S. Ahmed, and A. Zaqout, “Design and implementation of monitoring and control system for a poultry farm,” in *2020 International Conference on Promising Electronic Technologies (ICPET)*, pp. 44–49, 2020.

[13] D. Mancheno and S. Gamboa, “Development of an industrial iot gateway prototype,” in *Latest Advances in Electrical Engineering, and Electronics: Proceedings of the JIEE 2021*, pp. 61–75, Springer, 2022.

[14] H. Khujamatov, E. Reypnazarov, D. Khasanov, and N. Akhmedov, *IoT, IIoT, and Cyber-Physical Systems Integration*, pp. 31–50. Cham: Springer International Publishing, 2021.

[15] M. N. Elham, M. S. Sabeghi, F. S. Al-Rasbi, S. M. Daud, N. Q. Noor, R. Sahak, and A. A. Mazlan, “A preliminary study on poultry farm environmental monitoring using internet of things and blockchain technology,” in *2020 IEEE 10th Symposium on Computer Applications Industrial Electronics (ISCAIE)*, pp. 273–276, 2020.

[16] N. Watthanawisuth, A. Tuantranont, and T. Kerdcharoen, “Microclimate real-time monitoring based on zigbee sensor network,” in *SENSORS, 2009 IEEE*, pp. 1814–1818, 2009.

[17] G. A. Choukidar and N. Dawande, “Smart poultry farm automation and monitoring system,” in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1–5, 2017.

[18] M. S. Amin, S. T. H. Rizvi, U. Iftikhar, S. Malik, and Z. B. Faheem, “IoT based monitoring and control in smart farming,” in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pp. 1–6, 2021.

[19] L. Enciso and A. Vargas, “Interface with ubidots for a fire alarm system using WiFi,” in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, 2018.

[20] A. Roihan, F. Sudarto, and T. Cahyo Putro, “Internet of things on monitoring and control system in server area,” in *2018 International Seminar on Application for Technology of Information and Communication*, pp. 116–120, 2018.

[21] I. M. Reza Permana, M. Abdurohman, and A. G. Putrada, “Comparative analysis of mesh and star topologies in improving smart fire alarms,” in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, pp. 1–5, 2019.

[22] M. Ramos, “Characterization of LM35 Sensor for Temperature Sensing of Concrete,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2017 Vol II, IMECS 2017, Lecture Notes in Engineering and Computer Science*, (Hong Kong), pp. 760–764, 2017.

[23] S. Chatterjee, S. Chatterjee, and R. Gupta, “Arduino based real-time wireless temperature measurement system with GSM based annunciation,” in *2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0840–0844, 2017.

[24] I. Stoev, S. Zaharieva, A. Borodzhieva, and G. Staevska, “An approach for securing MQTT protocol in ESP8266 WiFi module,” in *2020 XI National Conference with International Participation (ELECTRONICA)*, pp. 1–4, 2020.

[25] J. Mesquita, D. Guimarães, C. Pereira, F. Santos, and L. Almeida, “Assessing the ESP8266 WiFi module for the Internet of Things,” in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 784–791, 2018.

[26] D. Papakyriakou and I. S. Barbounakis, “Benchmarking and review of Raspberry Pi (RPi) 2B vs RPi 3B vs RPi 3B+ vs RPi 4B (8GB),” *International Journal of Computer Applications*, vol. 975, p. 8887.

[27] T. Pan and Y. Zhu, *Designing Embedded Systems with Arduino*. Springer, 2018.

[28] E. Boonchieng, O. Chieochan, and A. Saokaew, “Smart farm: Applying the use of NodeMCU, IOT, NETPIE and LINE API for a Lingzhi mushroom farm in Thailand,” *IEICE Transactions on Communications*, vol. 101, no. 1, pp. 16–23, 2018.

[29] A. Zainudin, M. F. Syaifudin, and N. Syahroni, “Design and implementation of node gateway with MQTT and CoAP protocol for IoT applications,” in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 155–159, 2019.

[30] D. Ortiz, D. S. Benítez, W. Fuertes, and J. Torres, “On the use of low cost sensors for the implementation of a real-time air pollution monitoring system using wireless sensor networks,” in *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pp. 1–6, 2018.

[31] D. J. R. Enriquez, M. Lowell G. Villanueva, S. M. G. Dumlao, and R. S. Reyes, “Wireless power consumption monitoring and analysis system using winter’s forecasting method,” in *TENCON 2015 - 2015 IEEE Region 10 Conference*, pp. 1–6, 2015.

[32] A. Antony and S. Sarika, “A review on IoT operating systems,” *International Journal of Computer Applications*, vol. 176, pp. 33–40, 2020.

Appendix A

```python
from ubidots import ApiClient
import serial
import time

if __name__==’__main__’:
    dato=0
    try:
        print("Connecting...")
        arduino=serial.Serial(’/dev/ttyS3’, 9600)
        time.sleep(1)
    except:
        print("Fail to connect")
    try:
        print("Connecting API...")
        api=ApiClient(token=’tokenID’)
        temperature=api.get_variable(’ID1’)
        weight=api.get_variable(’ID2’)
        print(’testing connectivity’)
    except:
        print("Fail to connect API")

    while True:
        temp=arduino.readline()
        dato=float(temp)
        print("Temperature in Centigrade")
        print(dato)
        newTemp=temperature.save_value({’value’:dato})
        load=arduino.readline()
        c=float(load)
        print("Weight/Load in Kg")
        print(c)
        newLoad=weight.save_value({’value’:c})
        print("  ")
```

Listing 1: Coding for test connectivity and printing results