Monitoring a turkey hatchery based on a cyber-physical system

1st Anthony Maisincho-Jivaja (ID) , 2nd Ulbio Alejandro-Sanjines (ID) , 3rd Víctor Asanza (ID)
, 4th Tonny Toscano-Quiroga (ID) ,5th Nadia N. Sánchez-Pozo (ID) , 6th Leandro L. Lorente-Leyva (ID)
and 7th Diego Hernan Peluffo-Ordóñez (ID)

1,2,3,4Facultad de Ingeniería en Electricidad y Computación, FIEC
Escuela Superior Politécnica del Litoral, ESPOL
Campus Gustavo Galindo Km 30.5 Vía Perimetral, P.O. Box 09-01-5863
090150 Guayaquil, Ecuador
{amaisinc,ualejand,vasanza,ttoscano}@espol.edu.ec
3,5,6,7Smart Data Analysis Systems Group (SDAS Research Group - www.sdas-group.com),
Ben Guerir 47963, Morocco
{victor.asanza, nadia.sanchez, leandro.lorente, diego.peluffo}@sdas-group.com
7Modeling, Simulation and Data Analysis (MSDA) Research Program, Mohammed VI Polytechnic
University, Ben Guerir 47963, Morocco
peluffo.diego@um6p.ma

Abstract—**The implementation of a turkey farm brings with it severe environmental problems due to the deficient study of the physical space where the animals are placed. To counteract this situation and improve the quality of life in the hatchery, it is necessary to monitor and control the following variables: Temperature, Humidity, Ammonia Emission and Lux. The solution is based on a cyber-physical system which is composed of a network of sensors, controller and actuator. The sensors will provide information from the physical environment, the controller evaluates these parameters to execute an action to the actuator. Proportional, Integral and Derivative (PID) control defines the setpoint for temperature while Pulse-Width Modulation (PWM) adjusts the light intensity in a spotlight. The End Device executes these actions and its parameters will be sent to ThingSpeak which monitors system behavior the Internet of Things.**

Keywords—**Internet of Things, PID Tunner Toolbox, System Identification Toolbox, Meleagriculture, Sensor Network.**

I. INTRODUCTION

Turkey production is growing rapidly worldwide, with the American continent standing out, which together account for nearly 80% of the poultry census. In this sense, this increase has forced the implementation of industrialized and automated hatchery systems. Its objective is to improve health and biosafety by controlling the parameters involved in the growth phase of successful production [1].

There are difficulties in the production of this bird in Ecuador due to the inefficient hatcheries in the poultry production centers. Adapting the automated turkey farms used in first world countries would be extremely expensive and uneconomical [2].

Therefore, a cyber-physical system based on a wireless network of sensors and actuators is proposed. The sensors will take measurements with respect to the environment such as: temperature, humidity, carbon dioxide and ammonium in the environment.

The actuator will adjust the temperature to prevail in a standard range in the hatchery. It will have the ability to send the data to the cloud a coordinator, with low power consumption. The data will be visualized by ThingSpeak for constant access without the need to be on site. This will allow automated control over the variables of interest, thus improving preventive maintenance planning and turkey quality of life [3].

The rest of the paper is organized as follows. Section 2 provides an extensive overview of the related literature. Sections 3 describes the experimental setup performed in the data acquisition process and the methodology used in end device. Section 4 explains the experimental methodology for the gateway. Section 5 presents the results obtained from the sensors and the performance of PID control. Finally, the discussion of the results obtained is presented in section 6, followed by the conclusions.

II. RELATED RESEARCH

A. Network of XBEE modules for the control and the monitoring of temperature and light intensity for poultry farms through API mode with HMI visualization.

Torres et al., proposed as a solution the implementation of a wireless network prototype which permits in an interactive way making use of the variables and

adapting them to the requirements of the user who are the people in charge of production and raising of chickens in the sheds [4].

The aim is to design the network having as its primary mean the communication the XBee modules, which will transmit all the information extracted from the sensor circuits, and the data obtained will be sent to the computer through the human machine interface (HMI) without the need for cabling.

B. Development of an automatic feed dispenser implemented with Arduino UNO in poultry production.

The project comes from the importance of food quality for poultry production generated by the livestock sector. Because of this, the programming of sensors that indicates the temperature in the food is implemented with open hardware to enhance the health of the chicken in the coop. The main controller is the Arduino Uno board which will sense and provide the exact amount that the animals require for their feeding in order to have an automatic control since it reduces the cost of labor and uncleanliness in the chicken coop, thus avoiding feces or insects that can alter the quality of their feeding [5].

C. Design of a management control system for Poultry Plants using Wireless Sensor Networks with Open Hardware technology.

Open hardware technology was used for the implementation of sensors in a poultry plant. Open hardware technology offers the user the following: freedom of use, study and modification, distribution, and redistribution of improvements [6].

A microcontroller board was attached to another board which was specifically designed for the connection of sensors and actuators. The sensors were selected according to the technical characteristics of a broiler rearing process. They measure temperature, humidity, light intensity and ammonia, carbon monoxide and flammable gases levels.

The module sends the measurements using wireless devices with the ZigBee protocol to a computer which, using developed software, compares them with parameters introduced by the poultry farmer. It is of utmost importance that the data be constantly monitored since maintenance could be carried out in a poultry farm as well as risk prevention in a hydrometeorological system [7].

III. EXPERIMENTAL END DEVICE METHODOLOGY

The different End Devices used for the cyber-physical system are described below: The most important parameters involved in the growth phase are temperature, lighting, and the presence of ammonia

Table I
Description of the functions of each End Device.

Device	Function
End Device 1	Humidity and Temperature Measurement
End Device 2	Lux and Ammonia Measurement
End Device 3	Focus PID Control

Table II
Parameters and range of values for the turkey farm.

Parameter	Description	Range
Temperature 1	Under heat source	40°-45°
Temperature 2	Ambient	32°-34°
Relative humidity	Ambient	60%-70%
Luminous intensity	Recommended light in the shed	60-65 Lux
Ammonia	Ammonia levels	$\leq 10 ppm$

[2]. It was decided to monitor the variables based on the optimal ranges for an optimal growth phase.

With the selection of the parameters to be monitored and controlled, an investigation of the sensors available in the market that comply with the established ranges and are compatible with the Atmega328P microcontroller was made.

The cyber-physical system comprises four sensors and an actuator; hence, it must be chosen which variable we want to control according to a range recommended by experts in poultry production. A 55W halogen bulb type H1 is used, with a 12 VDC supply voltage. A PID control was implemented to regulate the power in the lamp through PWM pulses. These pulses change their duty cycle depending on the error between the reference, which will be 42 ° C, and the temperature under the heat source.

Table III
Sensores utilizados para el monitoreo.

Sensor	Magnitude	Measuring range
LM35	Temperature	From - 55° to 150°
LDR	LUX	Adjustable
MQ-137	Ammonia concentration	From 5 to 200 ppm
DHT11	Temperature and Humidity	From 20% to 92%

The sensors are distributed in the specific End Devices, the star network topology is implemented. Communication between the coordinator and the three end devices will be through the IEEE 802.15.4 Zigbee protocol. This communication protocol allows several XBee devices to establish networks as PEER-TO-PEER or POINT-TO-MULTIPOINT, reaching theoretical ranges of up to 45 km at full line-of-sight [4]. For the present project, the XBee PRO S1 was used, which works in the 2.4 GHz frequency band, with a data transmission of 250 kbps. The physical layer between

the Raspberry and the XBee antenna is through the Zigbee shield with USB port.

The End Devices will have in their Free RTOS code the reception of the frames sent by the Raspberry. The physical layer for the data flow with the coordinator will be USB through the XBee antennas. The coordinator requests information from the End Devices through a frame, receives and publishes data in ThingSpeak, which is available through the internet.
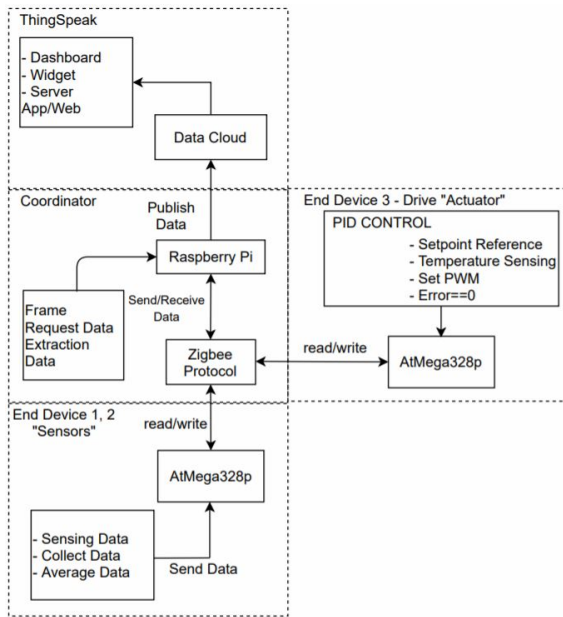

Figure 1. *Network topology used.*

Fig. 1 represents the topology applied, indicating that programming is performed on the end devices using FreeRTOS, which is an open source, low performance, real-time operating system, ideal for embedded system devices. A FreeRTOS library is applied in the Arduino IDE, which together with the XBee wireless communication module, will communicate with the coordinator through frames. For the first End Device, the readings from the DHT11 sensor will be sent by serial communication to the coordinator, however these data are sent with a frame, so that the first and the second data are recognized and that there are no errors in their reception. The following is the pseudocode

used for End Device 1 and 2.

Algorithm 1: Pseudocode End Device 1.

Result: Send average data every 15 seconds.
Enabled end device 1;
time=0;
while *time < 15s* **do**
    Accumulate temperature data;
    Accumulate humidity data;
    time+=1;
end
Send average temperature data;
Send average humidity data;

The Lux reading with respect to the LDR found in the End Device 2, is carried out using Eq. (1).

$$Lux = \frac{\frac{2500}{Vo} - 500}{R} \qquad (1)$$

Where, R represents the value of the resistance in series with the component, Vo is the voltage drop which the LDR will have, and Lux is the result of formula aforementioned, this is the unit of measurement which is monitored in IoT [8].

System Identification for End Device 3, to find the PID constants of the controller. The system identification allows us, through experimentation, to find the representation of the plant to control it [9]. Fig. 2 details the methodology of how the PID constants were obtained in our controller. That being the case, the time it takes for the temperature to stabilize when the light intensity is modified by means of the voltage which varied between 0 to 7.451 V (system input) was obtained in open loop, changing from 28.8 ° C to 62.98 ° C the temperature in the plant (System outlet). The period of the square wave generator is twice the largest of the sampling in open loop of 117316 samples, the sampling frequency is 100 Hz. Then the data, which can be found in the repository of IEEE-DataPort.org [10], were pre-processed to be entered in the MATLAB Toolbox System Identification that permits the identification of systems [11]. The data acquisition allowed to have 5 open-loop periods, which would serve to denote the 70 % of the signal as training and 30 % for validation. It is important to note that the 5 periods were not enough, therefore, the concatenation of the existing data was performed in order to have more analysis in the MATLAB mathematical models. In addition, in the post-acquisition, the sampling frequency was reduced to 0.83 Hz since there were repeated data for 120 samples, so a sampling period of 1.2 s per data was defined.

In it, an adjustment percentage of 81.34% was found between the calculated plant and the real one obtained from the data, which is an outstanding approximation

with a transfer wave function of one zero and two real poles. The Transfer function is shown in Eq. (2).

$$G(s) = \frac{0.01166s + 3.551e^{-0.5}}{s^2 + 0.1335s + 7.651e^{-0.5}} \quad (2)$$

Following that, the dynamics of the control system are adjusted by means of the Toolbox PID Tuner in MATLAB, where the information on the signal characteristic is provided to obtain the constants of the PID-type controller, which in this case was PI.

Table IV
PID constants and characteristics of the closed loop system.

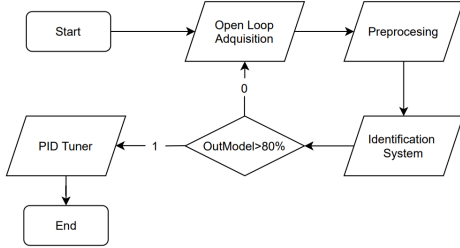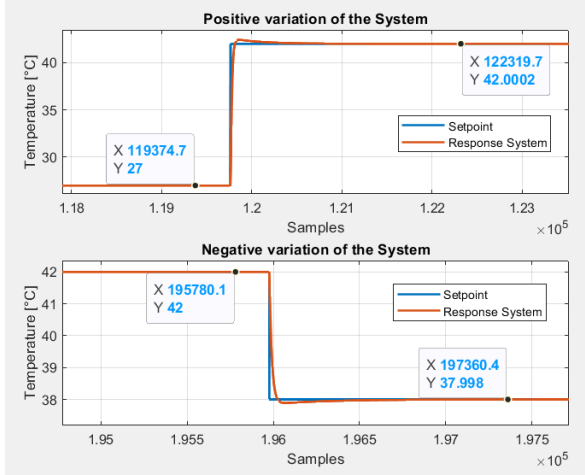| Controller Parameters		Performance and Robustness	
Kp	5.9843	Rise time	27.8 [s]
Ki	0.81832	Settling time	212[s]
Kd	0	Overshoot	2.77%
Tf	n/a	Peak	1.03
		Closed-loop stability	Stable

Figure 2. *Procedure for obtaining PID constants.*

Figure 3. *System Response with a Setpoint Adjustment.*

Fig. 3 shows the closed-loop behavior of the plant, with the PI constants obtained and as a function of the parameters introduced in the PID Tuner. These plots were obtained with the real system to identify the deficiencies that could occur in our controller. For this reason, small positive and negative variations were made around the operating point in temperature so as not to affect the linearity of the implemented system. This allows to reduce the error between the setpoint and the one obtained in the plant.

IV. EXPERIMENTAL GATEWAY METHODOLOGY

Fig. 4 details the functions and actions to be performed by the coordinator. The microprocessor that will have these responsibilities is the Raspberry Pi 4B, which will send a series of frames to the three end devices using the Zigbee protocol for wireless communication. These three frames will activate the readings of the different sensors of End Device 1 and 2, in addition to activating the PID control.

Every 15 seconds the data will be displayed in the ThingSpeak cloud, due to the free limitations of the API. As long as the data is not sent to the cloud, it will have an energy saving consumption for the system. Communication between the Coordinator and the End Devices is sent and received by the XBee PRO S1 antennas, which were configured for a maximum range of 1.1 km where the coordinator was located in a POINT-TO-MULTIPOINT network.
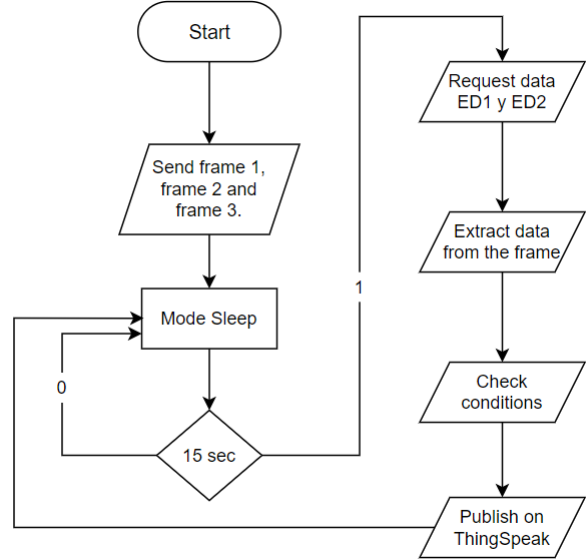

Figure 4. *Coordinator.py flowchart.*

On ThingSpeak, the data from the sensors were published with the intention that a constant monitoring of the parameters exists.

V. RESULTS

For the system simulation, the data from the sensors connected to the end devices are sent to ThingSpeak

through the coordinator, which comes out every 15 seconds due to its free version. Fig. 5, we have the temperature in the PID control, where it can be seen how it is adjusted and maintained at approximately 42 °C.
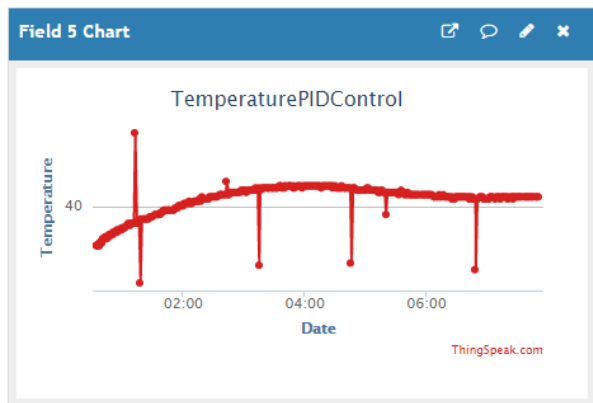

Figure 5. *Publishing records to ThingSpeak.*

To obtain data memory usage, or RAM memory used, in each of the End Devices, the different codes that were written with FreeRTOS were executed and in this way the following data was collected.

Table V
Data memory and SRAM usage for each end device.

End Device	Data Memory%	SRAM%
End Device 1	41%	19%
End Device 2	40%	19%
End Device 3	42%	21%

Furthermore, it is of utmost importance to consider the power consumption of our cyber-physical system. In the case of End Device, as it is constantly making analog and digital conversions, it is safe to say that its consumption will be constant. In the case of the coordinator, which is the Raspberry PI 4B, it usually has a consumption of 4.1 W, however, for 15 seconds it goes into sleep or "idle" mode, consuming 2.7 W. It is as well necessary to take into account the consumption of the Xbee modules, which will act every 15 seconds.

Table VI
Consumption of used devices.

Device	Vin	Current	Power
End Device 1	5v	13.5mA	67.5mW
End Device 2	5v	11.06mA	55.3mW
End Device 3	5v	200mA	1W
Xbee Pro S2	5v	50mA	25mW
Raspberry Pi(Active)	5v	540mA	2.7W

VI. Discussion and Conclusions

The project described above solves the main defects that occur in poultry farms in Ecuador. The use of AVR microcontrollers such as the Atmega328P and hardware based on ARM Cortex A72 processor such as the Raspberry Pi 4B, working with low power consumption and with maximum savings in downtime or IDLE operation mode. At the same time its price becomes more competitive because of all the features offered by the system, such as its wireless sensor network, sending data to the cloud, user-defined temperature control and its maximum wireless connection range of 1.1 km between devices.

By means of the MATLAB Toolbox System Identification, the behavior of the plant was obtained experimentally. By reducing the sampling frequency in the post-acquisition of data through code, we avoided repeating the open-loop experiment. Recall that the temperature is a slow variable to reach from its operating point to the steady state, so it was better to reduce the sampling frequency by code in MATLAB than to repeat the physical experiment.

The memory usage in each of the End Devices varies depending on the tasks assigned. In this case, when using the FreeRTOS library, the programming consumes more computational resources, because it is necessary to load a boatloader and assign stacks for each task prior to the use of the library [12]. However, for the programmer, being a language that has a lower learning curve, he could adapt his project to new needs with less development time.

Regarding energy consumption, sending and publishing data to the cloud every 15 seconds is an effective way to save energy, because there is half the consumption in downtime, it is important to take into account that variables such as humidity or ammonia levels are slower variables and can be sent with a longer sampling period, optimizing energy consumption in the Raspberry Pi.

The use of the PI controller embedded in the Atmega328P microcontroller, allows the system to have a better response time control of the temperature variable. This is very important for the end user, since he will be able to make adjustments and reach the reference temperature with high precision and an adequate stabilization time to maintain the temperature in the hatchery.

VII. Future Work

The End devices, together with the coordinator, could be replaced by a LoRa ESP32, becoming a lower cost project with new advantages such as long-range data transmission, low consumption, and direct connection to the Internet via Wi-Fi. These are programmed through the arduino IDE, using a program-

ming language that is easy to understand and takes less development time.

The Raspberry Pi Pico can be implemented by the Arduino for data acquisition in the system, as well as its monitoring of the variables. Which is much more economical because it considers a Dual Core ARM Cortex processor, SoC system of chip. In addition to having more Flash and Ram memory than in an Arduino nano, improving performance in their applications. Another advantage of using the Raspberry Pi Pico, is that no additional libraries are needed for multitasking, saving program memory.

Regarding the Antenna, it is possible to improve the gain or signal power, in order to reach greater distances. The system could consume more power and keep better characteristics with respect to data transmission and reception. It is possible to use omni-directional antennas to have greater coverage between points of interest.

For greater precision in the measurement of lighting levels in the shed, you could choose to change the LDR sensor for the BH1750, which gives us the light intensity directly in Lux [Lx] with less code development. Furthermore, it is important to control variables such as ammonia by activating an extractor.

REFERENCES

[1] Santos-Ricalde, R., Segura-Correa, J. (2020). La importancia de la cría de pavos (Meleagricultura) en Yucatán. Bioagrociencias, 13(1).

[2] Marchewka, J., Watanabe, T. T. N., Ferrante, V., Estevez, I. (2013). Review of the social and environmental factors affecting the behavior and welfare of turkeys (Meleagris gallopavo). Poultry Science, 92(6), 1467-1473.

[3] Noll, S. L., El Halawani, M. E., Waibel, P. E., Redig, P., Janni, K. (1991). Effect of diet and population density on male turkeys under various environmental conditions.: 1. Turkey growth and health performance. Poultry science, 70(4), 923-934.

[4] Torres Sánchez, R. D. (2019). Red de módulos XBee para el control y monitoreo de temperatura e intensidad luminosa para criaderos avícolas mediante modo API con visualización HMI (Bachelor's thesis, Quito).

[5] Soh, Z. H., Ismail, M. H., Otthaman, F. H., Safie, M. K., Zukri, M. A., Abdullah, S. A. (2017, November). Development of automatic chicken feeder using arduino uno. In 2017 International Conference on Electrical, Electronics and System Engineering (ICEESE) (pp. 120-124). IEEE.

[6] Barrera, F. V., Granja, F. M. (2017). Diseño de un sistema de control gerencial de plantas avícolas utilizando redes de sensores inalámbricos con tecnología Open Hardware/Design of a management control system for poultry plants using wireless sensor networks with Open Hardware technology. Ciencia Unemi, 9(21), 143-156.

[7] Romero, G., Salazar, C., Asanza, V. (2015). Desarrollo de un Prototipo de Sistema Hidrometeorológico. Revista Tecnológica-ESPOL, 28(5).

[8] Hapidin, D. A., Hernawan, M. Z. P., Krisnanto, F., Syahbana, A., Fiordi, M. I., Munir, M. M., Khairurrijal, K. (2018, December). The Study of Velocity Measurement Using Single Light Dependent Resistor (LDR) Sensor. In 2018 3rd International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM) (pp. 111-114). IEEE.

[9] Ljung, L., Singh, R. (2012). Version 8 of the MATLAB system identification toolbox. IFAC Proceedings Volumes, 45(16), 1826-1831.

[10] Anthony Maisincho J, Ulbio Alejandro S, Víctor Asanza, Tonny Toscano Q, March 21, 2021, Temperature data acquisition for a turkey hatchery. IEEE Dataport, doi: https://dx.doi.org/10.21227/hsqh-5959.

[11] C. Angel Silva, V. Asanza, N. Sánchez and J. Arias, "Implementation Of Automated System For The Reservoir 66 of the Irrigation System Chambo Guano," 2020 International Conference on Applied Electronics (AE), 2020, pp. 1-6, doi: 10.23919/AE49394.2020.9232888.

[12] V. A. Armijos, N. S. Chan, R. Saquicela and L. M. Lopez, "Monitoring of system memory usage embedded in FPGA," 2020 International Conference on Applied Electronics (AE), 2020, pp. 1-4, doi: 10.23919/AE49394.2020.9232863.