Raspberry Pi-based IoT for shrimp farms Real-time remote monitoring with automated system

1st Jesús Capelo, 2nd Erick Ruiz, 3rd Víctor Asanza, 4th Tonny Toscano-Quiroga, 5th Nadia N. Sánchez-Pozo,
6th Leandro L. Lorente-Leyva and 7th Diego Hernan Peluffo-Ordóñez
[1,2]*Facultad de Ingeniería Mecánica y Ciencias de la Producción, FIMCP*
[3,4]*Facultad de Ingeniería en Electricidad y Computación, FIEC*
Escuela Superior Politécnica del Litoral, ESPOL
Campus Gustavo Galindo Km 30.5 Vía Perimetral, 090150 Guayaquil, Ecuador
{jesfacap, eriaruiz, vasanza, ttoscano}@espol.edu.ec
[3,5,6,7]*Smart 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
[7]*Modeling, Simulation and Data Analysis (MSDA) Research Program,*
Mohammed VI Polytechnic University, Ben Guerir 47963, Morocco
peluffo.diego@um6p.ma

***Abstract*—This project analyses the optimal parameters for the *shrimp farming*, trying to help the aquaculture of *Ecuador*, using a *cyberphysical system*, which includes *temperature*, *salinity*, *dissolved oxygen*, and *pH* sensors to monitor the water conditions and an embedded system to control it using an *XBee* and *ATMega328p* microcontrollers to remotely activate and deactivate *aerators* to maintain the quality of each pool in neat conditions.**

Keywords — shrimp farming, cyberphysical system, Ecuador, XBee, ATMega328p, temperature, salinity, dissolved oxygen, pH, aerator

I. INTRODUCTION

For shrimp rearing, one of the most important factors to consider is the quality of the water, because being this the environment in which shrimp are developed directly influences the health of the animal, as well as its growth and reproduction factor, being a critical variable for both industry and the environment [1].

Having a healthy shrimp, is important if you want to obtain a good harvest, for which it is important to monitor its temperature, since studies have confirmed that, if the temperature is outside a permitted range, the shrimp may suffer from stress processes or some other disease, which can undermine its breeding process, or cause it to be discarded at the time of collection, thus reducing its production [2].

This is important to keep in mind since after bananas, shrimp is the second most exported non-oil product in our country, so it is important to carry out a correct control of the product, in such a way that it seeks to increase its production and quality exponentially, allowing to improve and exploit the international market [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 efficiency of the End Device. Finally, the discussion of the results obtained is presented in section 6, followed by the conclusions.

II. RELATED WORK

Wardhany et Al. focuses on the importance of salinity, temperature, and pH control, for shrimp survival, while setting parameters of these values, in which the product has a growth, for this they use a microcontroller responsible for receiving this signal and taking actions that allow to always keep these parameters within the correct margin. In addition, they implemented a data processing system using fuzzy logic, by which categories are created for each measured parameter [1].

On the other hand, a study conducted by Al Azzani AAH, aims to change the traditional monitoring system in shrimp pools, increasing efficiency and reducing human work and time in the area. To accomplish this goal, was implemented as IoT monitoring system, by using a Raspberry Pi and an Arduino Uno, that send the information to a database that in turn allows remote control and monitoring, from an application installed on a mobile device. These parameters to be monitored are temperature, pH of water, percentage of oxygen dissolved in water, and percentage of NH_3. On the other hand, an engine also controlled from the application makes the functions of aerator of the crop [2].

At the same time, a study conducted by Rerkratn, A., & Kaewpoonsuk, A., used the benefits of LPWAN technology, so it is decided to use Zigbee technology to interconnect all pools. Thus, through sensors connected to Xbee Pro modules connected to an Arduino Uno, which is in turn connected to a control program in LabView, water temperature control is achieved on the shrimp farm [4].

Another work carried out by Zainuddin et Al. they taked as parameters to be monitored, the pH of the water, the temperature at which it is located, and the turbidity of it. Then, all the information obtained is processed in an Arduino, which sends the information to an XBee module that then processes it to another receiving module, which also connects to an Arduino and thus displays the information on an LCD

Supported by Escuela Superior Politécnica del Litoral.

2021 International Conference on Applied Electronics (AE) | 978-80-261-0973-0/21/$31.00 ©2021 IEEE | DOI: 10.23919/AE51540.2021.9542907

ISBN 978-80-261-0973-0, © University of West Bohemia, 2021

screen. In addition, an ESP8266 module is used to connect this information to the internet, using a database [5].

Other authors, such as Sneha, P.S., & Rakesh, V.S,propose a real-time monitoring and control system for shrimp pools, for this purpose a system embedded in the pools is used, consisting of an Arduino Uno microcontroller, which will be responsible for monitoring and sending in real time the information collected from all sensors to a Raspberry Pi, which will process the information to decide if it is necessary to activate any actuator. In addition, this system has the benefit of IoT technologies, which will be responsible for sending the information to the internet, to be able to do things like control the system through a mobile device and the Telegram application, or have an SMS alert system, which allows the user to always know if there is any urgency in their shrimp [6].

From the jobs previously described a combination between technologies used in all of them is used, to monitor a control the system. Therefore, this job arises the development of an IoT device, combining the Low Power Wide Area Network (LPWAN) a control based on hysteresis, to monitor several parameters of water quality, having a decent scalability by using one Raspberry Pi for each pool, connected via Serial Communication to an Atmega328p and a Xbee, to use connect the whole farm to the internet and control it remotely.

III. EXPERIMENTAL METHODOLOGY

Once the research was carried out, we continue to find out the crucial parameters needed by shrimp within shrimp farms, where research conducted by Al Azzani AAH [2] showed the following:

Table I. Range of accepted values for the optimum shrimp farming.

Parameter	Minimum value	Maximum value
Dissolved oxygen	5 mg/L	10 mg/L
Temperature	20°C	30°C
Salinity	15 ppm	25 ppm
pH	6.5	8.5

Once this information was collected, we proceeded to search for the sensors and actuators necessary to be able to carry out the work, for which it was decided to choose the following sensors that can be seen in Table II.

Table II. Selected sensors for the measurement process

Sensor	Measurement	Range	Tolerance
LM-35	Temperature	-55 a 150 °C	±0.5
SEN0237-a	Dissolved oxygen	0 a 20 mg/L	±0.04
PH-BTA	pH	0 a 14	±0.02
DRF0300	Conductivity	5 a 20 mS/cm	±0.02

Following the choice of sensors, it was decided which of the variables shown in Table I was the most important to automate, which chose to use dissolved oxygen in water., [7] since it has been proven to have direct effects on the growth and reproduction stages of shrimp, [1] for which a DC Z2D30-24A engine was chosen, which would allow us to perform the oxygenation process of the water when it reaches levels below what is desired.

Having a model with four sensors and an actuator, the next step was to choose the model to be used for the End Device and the coordinator, and it was decided to use a Mixed Star / Tree Topology model, in which the End Devices are connected between nodes and are arranged in the form of a tree, and the distribution of information from the coordinator goes from a central point to the various nodes of the network (End Device) where each End Device will have its respective sensors and actuators, and the coordinator is in charge of receiving the information from each End Device and uploading it to the database, this type of topology is used for its efficiency and simplicity, it gives us the advantage that if a node fails, the network continues to work without inconvenience. and this will only depend on the functioning of the coordinator. Fig. 1 shows the proposed topology.

After the choice of Network Topology, we program the End Devices and coordinator, through which the desired control is achieved, for which an ATMEGA328p is chosen as the End Device (ED) next to an XBee, and a Raspberry Pi 3 is used as the Coordinator [8].

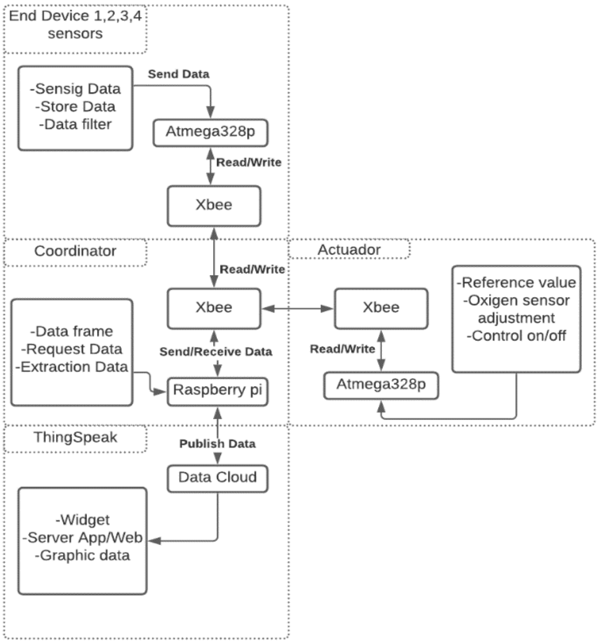

Figure 1. Network topologies used in this project.

The Fig. 2 shows the pseudo-code that indicates the steps to be able to create the program that can read in the End Device the analog values delivered by each of the sensors, through the use of an ADC that has the Atmega328 microcontroller where It has 8 input channels and these can be used to read the analog value that is within the reference voltage range, and that allows ON / OFF control for the necessary actuators.

Additionally, to send the information from the End Devices to the Coordinator, serial communication using frames was used, as can be seen in Fig. 3. that are sent and received by the coordinator, who performs the necessary conversions to transform a value from 0 to 5, to the values in the corresponding units of each sensor.

Once frames arrive from the End Devices to the Coordinator, the value received from the oxygen sensor is compared, so that if it is less than 7 mg/L, activate the On/Off control using a frame, in turn, it will activate a transistor., [9] which will supply the necessary power for the actuator to operate, until the oxygen sensor yields a value of 10 mg/L, so that the coordinator will send another frame responsible for shutting down the engine.

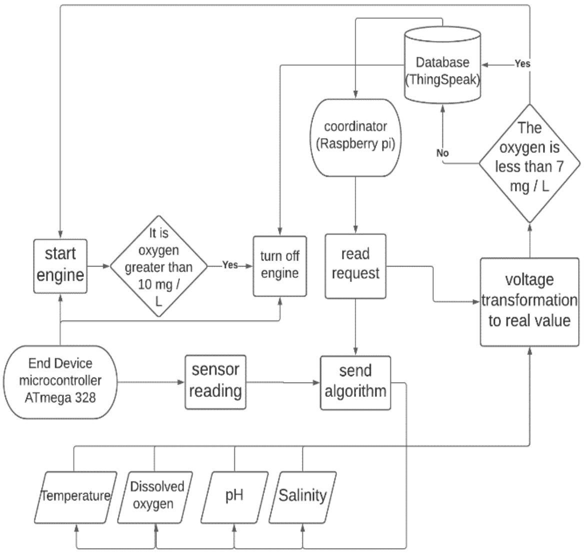
Database (ThingSpeak)
Yes
coordinator (Raspberry pi)
No
The oxygen is less than 7 mg / L
start engine
It is oxygen greater than 10 mg / L
Yes
turn off engine
read request
voltage transformation to real value
End Device microcontroller ATmega 328
sensor reading
send algorithm
Temperature
Dissolved oxygen
pH
Salinity

Figure 2. Gateway's Algorithm

The proposed circuit shown in Fig. 4, works at 24V, since it is the necessary value for the activation and correct operation of the actuators, however, it consists of some voltage regulators, which modify the voltage from 24V to 12V in order to escalate the voltage, to use any kind of additional actuators, and 5V, which is the main voltage used for the Raspberry Pi and the Atmega328p, which sends a signal pulse to an optocoupler which is a device that acts as a switch, the base of the transistor is connected to the optocoupler output that helps 24V switching, the emitter and collector are directly connected to the power source. power supply, so that it is activated when it receives the signal from the Atmega328p, allowing the actuators to be activated as well. Also, this level is used to maintain a constant or adjustable voltage level that feeds the different sensors found in Table II. Finally, a 3.3V regulator is used, so that it can feed the XBee-pro for communication to the LPWAN network [10].

```
Algorithm 1: Frame send (oxygen sensor)
  Result: Actuator activation
  Start
  AD1=0;
  Actuator on=(data frame described activation);
  Actuator off=(data frame described deactivation);
  Data[]=(average of sensed data);
  Data frame reading
  Data frame= received Serial port;
  while (Data frame == (Data frame described)) do
      AD1=sensor data;
      Data[]=AD1;
      (send AD1 by serial port);
      if ($AD1 \leq 7$) then
          Actuator on=(plot that activates the actuator);
      else $AD1 \geq 10$
          Actuator off= (plot that disables the actuator);
      end
  end
```

Figure 3. Algorithm's Pseudocode

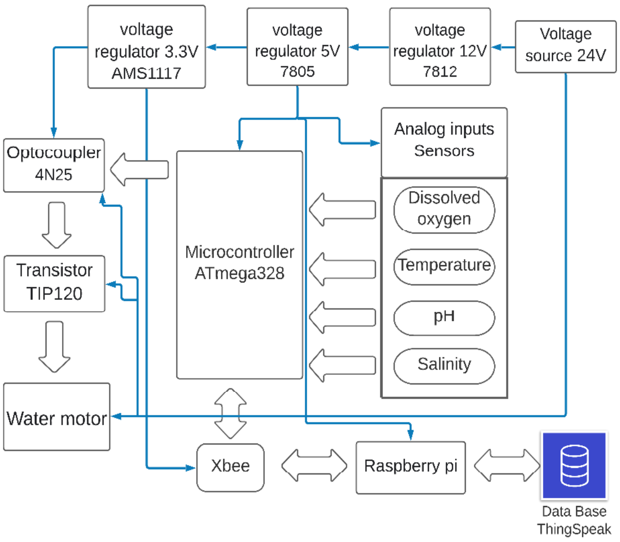

Figure 4. Electronic circuit of the Project described.

IV. RESULTS

To make an effective comparison of what is the best configuration to have a good profit/expense ratio, with this, we create the following configurations, where you can appreciate the number of microcontrollers per shrimp pool, in addition to the effective time in which you measure the respective values and the time you spend at rest, being reflected in the following Table III.

Table III. Possible configurations for the solution.

Conf.	# of End Devices by pool	Time in between plots	Idle Time
1	1	2s	90s
2	4	1s	60s
3	2	5s	90s
4	1	1s	45s

Having created our configurations, it is important to obtain the average consumptions that each of the components to use to calculate the energy expenditure. So, we know that a Raspberry Pi 3 running at 3.3 V requires a current of 25mA when idle, 250 mA when it is performing a process and an additional 16mA for each GPIO. On the other hand, an xBee-Pro S2, to receive frames needs from 35mA at 3.3V, while, to send frames, it requires a current of 232mA on average. Finally, an ATMEGA328p, operating at 3.3V, has a current of only 1.5mA [11].

Table IV. Power consumed by all devices.

Device	Voltage [V]	Current [mA]	Power [W]
Raspberry Pi 3 Idle	5	25	0.125
Raspberry Pi 3 Active	5	250	1.25
Raspberry Pi 3 GPIO	5	16	0.08
XBee-Pro S2 receiving plots	3.3	35	0.1155
Xbee-Pro S2 sending plots	3.3	232	0.7656
Xbee-Pro S2 idle	3.3	15	0.0495
ATMEGA328p	5	1.5	0.0075

With all the information collected, together with the settings shown in Table III, we proceed to make a calculation that allows us to know the power required by each of the devices when performing an action, for which we will use the Table IV. Having the information in Table IV, we proceed to calculate the power used per second in each of the configurations, considering downtime, such as time at which the devices are idle, and as time between frames, time in which the XBee-Pro S2 is inactive, these results are shown in Figure 5.

V. CONCLUSION

From Figure 5, the best configurations in terms of energy expenditure vs time are configuration 1 and configuration 4, this is because using a single End Device per pool saves the process of sending frames for each XBee used, a process that requires a lot of energy to be performed, compared to others. Through this analysis, it is possible to consider the importance factor of each of the parameters used for the configuration.

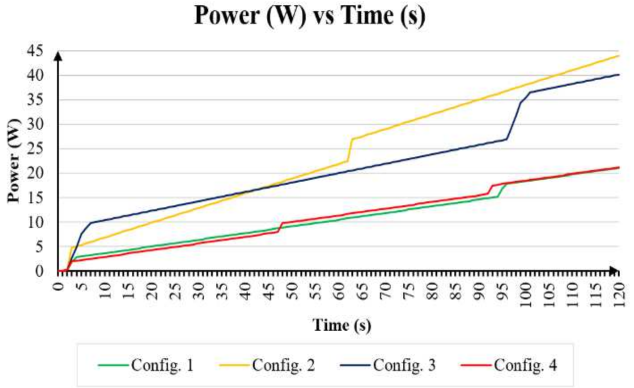

Figure 5. Power vs Time graphic comparing each configuration.

Thus, the second most important factor would be the downtime used by each sensor, which is why the best possible configuration would be the first, getting after 120 seconds a total consumption of 21.04 W, followed by the fourth, which gives us a consumption of 21.18 W, then a little higher are the third with a consumption of 40.11 W and finally the second with a consumption of 43.98 W.

These collected values are important to consider for the application, since depending on the need you can use a configuration with more measurement efficiency or one with more energy efficiency, for this case since all the measured variables of the system, are slow change, configuration 1 is sufficient for monitoring.

However, it is also important to analyze what is the best configuration in terms of efficacy and efficiency, so that it can be a standard model for any application. Therefore, based on the initial configurations and the results obtained, configuration 4 is recommended, since it ceases much faster than the configuration with less power consumption, without spending as much power as others.

Finally, it is important to consider as future work, in the medium term, to carry out a PCB design corresponding to this study, as well as to focus on the importance of new microcontroller technologies, such as an ESP32, which allows networking much easier than an ATMEGA328p. It is also important to emphasize the possibility of using IoT technologies for large areas of work with LPWAN networks, so that, through a database, it is possible to access and act according to information, without the need to be present, allowing remote monitoring and control.

VI. REFERENCES

[1] Wardhany, V. A., Yuliandoko, H., & AR, M. U. H. (2018, October). Fuzzy Logic Based Control System Temperature, pH and Water Salinity on Vanammei Shrimp Ponds. In 2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA) (pp. 145-149). IEEE.

[2] AL AZZANI, A. A. H. IOT BASED AUTOMATED SHRIMP FARM AQUACULTURE SYSTEM.

[3] Oviedo Alvarado, I. D., & Altamirano Flores, C. A. (2017). Telecontrol de sistema de alimentación automático para cultivo de camarón (Bachelor's thesis, Espol).

[4] Rerkratn, A., & Kaewpoonsuk, A. (2015, October). ZigBee based wireless temperature monitoring system for shrimp farm. In 2015 15th International Conference on Control, Automation and Systems (ICCAS) (pp. 428-431). IEEE.

[5] Zainuddin, Z., Idris, R., & Azis, A. (2019, March). Water Quality Monitoring System for Vannamae Shrimp Cultivation Based on Wireless Sensor Network In Taipa, Mappakasunggu District, Takalar. In First International Conference on Materials Engineering and Management-Engineering Section (ICMEMe 2018) (pp. 89-92). Atlantis Press.

[6] Sneha, P. S., & Rakesh, V. S. (2017, November). Automatic monitoring and control of shrimp aquaculture and paddy field based on embedded system and IoT. In 2017 International Conference on Inventive Computing and Informatics (ICICI) (pp. 1085-1089). IEEE.

[7] Novianto, D., Prajoko, S., Setiyowati, I., & Purnomo, E. (2020, April). Calibration of pH and oxygen sensors applied to aquaponic system. In Journal of Physics: Conference Series (Vol. 1517, No. 1, p. 012105). IOP Publishing.

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

[9] Pullano, S. A., Tasneem, N. T., Mahbub, I., Shamsir, S., Greco, M., Islam, S. K., & Fiorillo, A. S. (2019). Deep submicron EGFET based on transistor association technique for chemical sensing. Sensors, 19(5), 1063.

[10] Saha, H., Mandal, S., Mitra, S., Banerjee, S., & Saha, U. (2017). Comparative performance analysis between nRF24L01+ and XBEE ZB module based wireless ad-hoc networks. International Journal of Computer Network and Information Security, 9(7), 36.

[11] Serbanescu, M., Placinta, V. M., Hutanu, O. E., & Ravariu, C. (2017, March). Smart, low power, wearable multi-sensor data acquisition system for environmental monitoring. In 2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE) (pp. 118-123). IEEE.

[12] Cordova, R., Garcia, I. M., Munoz-Arcentales, A., Asanza, V., & Vargas, W. A. V. (2018). Modelo de red de comunicación que soporta un sistema de detección basado en criterios de robustez. IEEE Latin America Transactions, 16(10), 2600-2608.

[13] Mekonnen, T., Komu, M., Morabito, R., Kauppinen, T., Harjula, E., Koskela, T., & Ylianttila, M. (2017). Energy consumption analysis ofedge orchestrated virtualized wireless multimedia sensor networks. IEEE Access, 6, 5090-5100.

[14] Vigneswaran, D., Ayyanar, N., Sharma, M., Sumathi, M., Rajan, M., & Porsezian, K. (2018). Salinity sensor using photonic crystal fiber. Sensors and Actuators A: Physical, 269, 22-28.

[15] Nasution, T. H., & Harahap, L. A. (2020, September). Predict the Percentage Error of LM35 Temperature Sensor Readings using Simple Linear Regression Analysis. In 2020 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM) (pp. 242-245). IEEE.