Remote Control and Monitoring of Fish Farms using Wireless Sensor Networks

H. F. Ragai*, Ihab Adly‡, Hossam E. M. Sayour ** and Samuel Wilson¶
* Electronics and Communications Department, Faculty of Engineering, Ain Shams University
‡Tegrom, Cairo, Egypt
**Biomedical Chemistry Unit, Chemistry Dept., Animal Health Research Institute, 12618 Dokki, Giza, Cairo, Egypt
¶Computer and Systems Department, Faculty of Engineering, Ain Shams University

***Abstract*—This work represents a proof-of-concept for remote monitoring and control of fish farms using WSN (Wireless Sensor Networks). The system allows a 24/7 live monitoring of water quality (mainly Temperature, pH value and Dissolved Oxygen value) in fish farms. The system is remotely accessed through the Internet via Desktop PCs and Mobile Phones. The system has the capability of sending alarms via different methods (emails or website notifications) to fish farms administrators. Thus, allowing them to take appropriate measures by means of remote actuation of different utilities installed in the fish tanks. The system aims to reduce the accidental mortality of fish which might happen due to water pollution or sudden environmental changes.**

***Index Terms*—ZigBee; Wireless Sensor Networks; Aquaculture; Remote monitoring; pH measurement; Remote actuation**

I. Introduction

Water quality is very essential for survival and growth of aquatic organisms, and is a major concern for fish farms. maintaining water quality level within acceptable ranges in order for various aquatic organisms to carry out their life functions is addressed in published papers [1][2]. Water quality is determined by variables like temperature, Dissolved Oxygen, pH value, conductivity, concentration of ammonia and many others. Continuous monitoring of those variables is vital, and any deviation from the acceptable ranges will heavily affect the health of aquatic organisms.

Many fish farms in Egypt today still rely on outdated methods to measure the water quality in ponds, thus endangering the lives of aquatic organisms. Those obsolete methods resulted in major mass fish mortality which will be discussed later.

Monitoring water quality using WSN has been a point of interest as discussed in [3][4]. Previous implementations of such systems are discussed in [5][6]. At present there are mainly four methods for monitoring water environments, each of which has its advantages and disadvantages [8]:

1) Artificial sampling with portable water quality detecting devices and subsequent lab analysis. This method applies only to samplings on cross-sections of river and lakes with a sampling frequency ranging from several times a day to monthly.
2) Automatic and continuous monitoring of water environment parameters by a system consisting of monitors and control centers, as well as several monitoring sub-stations. Data can be remotely and automatically transferred. Each station provides its real-time water environment parameters. These systems can be costly and have a great influence on the surrounding ecological environment.
3) Water environment monitoring with remote sensing technology, namely detecting the spectrum specifics of an electromagnetic wave (radiation, reflection and scattering) in a non-contacting method with respect to the water body. After the processing of the information from the collection of illustrative spectra, its physics and chemical characteristics are to be identified. However this method can only provide a low accuracy, and it is also hard to perform real-time monitoring.
4) Water quality monitoring technology realized using some sensitivity of aquatic organisms to the presence of poisonous substances in water bodies by measuring or analyzing the change of activities of different organisms in different water environments, then coming to a qualitative evaluation report of the water quality. Basic measuring methods of this type being practiced include Fish Measuring and Beach Louse Measuring. Still, these methods can by no means be expected to reach high accuracy for water environment monitoring.

Many of the above methods' setbacks can be tackled by implementing Wireless Sensor Networks (WSNs) which by comparison are cheaper and more convenient to use, which in terms offer real-time monitoring of all the needed variables and immediate actuation upon any deviation from the safe ranges [5] [6].

This paper aims to describe and develop a cross-platform system for advanced Aquaculture real-time monitoring and remote-control using ZigBee based WSNs. The system is based on Python, Websockets technology, SQL and PHP.

Further details about the system implementation and features are covered in the *System Architecture* section.

II. Motivation for a WSN Fish Farm System

Traditional fish farms do not implement any means of water quality measurement. As a result, a severe degradation in health occurs for the aquatic organisms which, in most cases, leads to death.

978-1-5386-1191-3/17/$31.00 ©2017 IEEE

An incident that took place in Singapore, where hundreds of tonnes of fish - both farmed and wild - died over the weekend is a clear example of the need for continuous monitoring and immediate actuation. The environmental authorities stated that the deaths were due to a plankton bloom, where a species of plankton multiplies rapidly, damaging the gills of fish. It can be triggered by sudden changes in temperature, high nutrient levels in the water, and poor water circulation [9].

Another incident that took place in Indonesia, where the mass mortality of the fish in Maninjau lake is apparently caused by an upwelling, with Sulfur-rich colder water from the bottom rising to the surface due to a drastic weather change, according to fishery officials. The same incidence occurred before in 2009 causing the death of at least 7,000 tonnes of fish [10].

It is needless to say that such incidents could not have been occurred if not for poor monitoring. However, traditional fish farms still prefer to use the outdated methods of farming as implementing a new system might seem intimidating. All the mentioned factors can be overcome by either replacing the current outdated methods with and/or adding WSN topology.

III. Literature Review

Multiple published papers [11], [12], [13] have discussed the need for wireless sensor networks for monitoring and/or remote-controlling in aquaculture.

The mentioned systems proposed using ZigBee protocol for low power consumption [13] and battery-powered wireless sensor node [12] for short distance communication or using GSM Modules [11] for environments where the base stations is kept distant from the sensor nodes. The idea of using ZigBee and/or GSM/CDMA originated after considering the main disadvantages of the currently used wire and cable for data communications such as high installation cost of communication and power supply for the sensors, difficulty in the installation, sensor data distortions due to temperature changes on cables, noise affecting cables and sensors etc. Most proposed systems require having a GUI application installed on Windows Platform [11], [12]and/or Android application [13] in order for the system to function; meaning they are platform-dependent.

From all the aforementioned related works, the use of ZigBee [7] in WSNs adds countless possibilities and can be deployed on various platforms according to the user requirements.

IV. System Architecture

Fig. 1 shows the overall architecture of the proposed system. The user interacts with a website which was implemented using the responsive design technology that enables it to work efficiently across various devices with different screen resolutions. The website opens a websocket connection with a python script that resides on the server and has direct connection with the ZigBee Coordinator (ZC), the data is then exchanged from the MySQL database and sensor nodes (via ZigBee End Device), and then pushed back to users. The

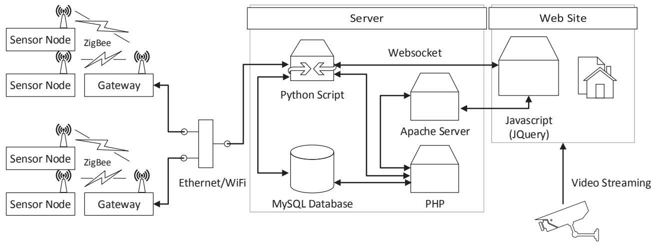


Fig. 1: System diagram.

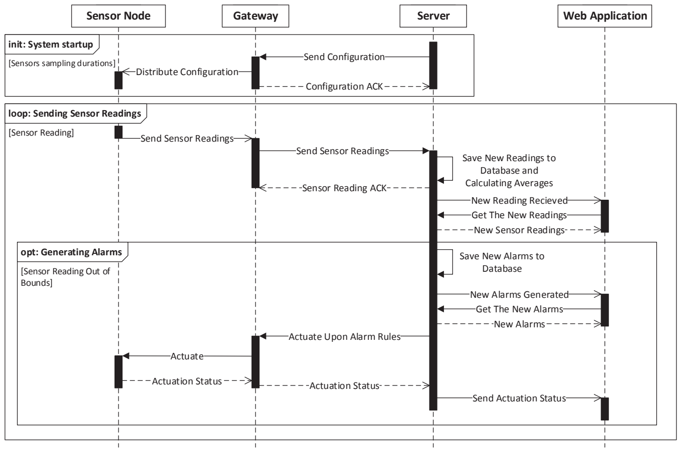
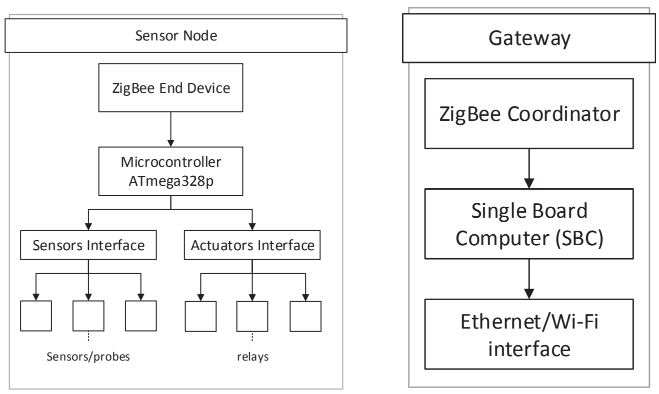


Fig. 2: Sequence diagram.

existence of PHP adds user authentication and other features which will be discussed in details shortly.

Fig. 2 shows the overall normal sequence of the system. Upon start the server will send a configuration file to each gateway, then the gateway will direct each configuration to its corresponding sensor node. The configuration file send contains the required sampling duration of each parameter and state of actuators (on/off).



(a) (b)

Fig. 3: Block Diagram of Sensor Node and Gateway

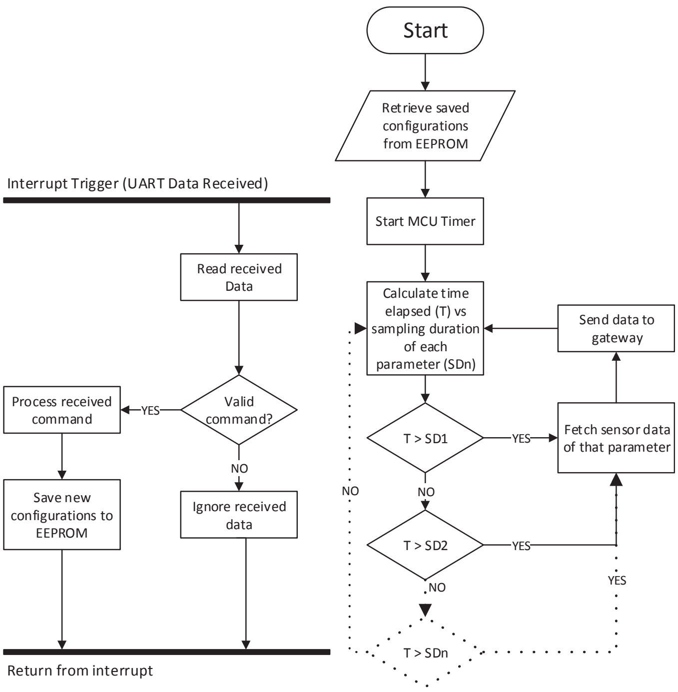


Fig. 4: Simplified flowchart of the C++ code deployed on the sensor nodes

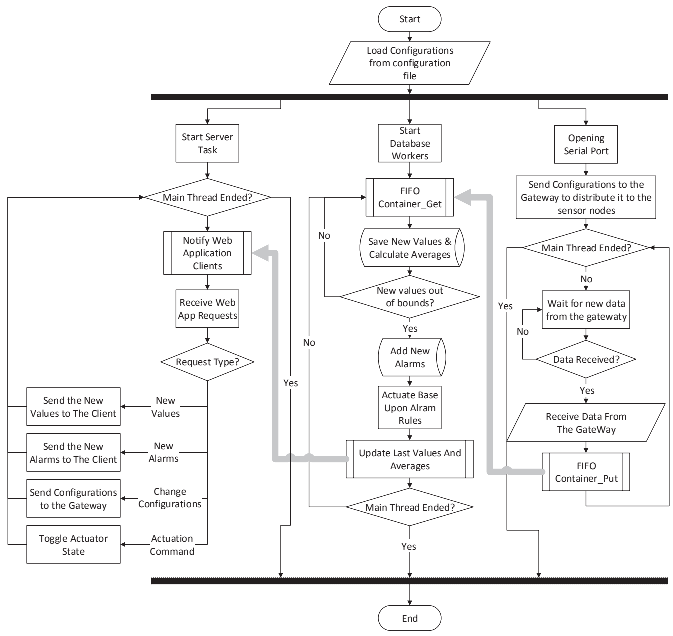


Fig. 5: Simplified flowchart of the python script deployed on the server

TABLE I: User tiers privileges

Features	**Administrator**	**Operator**	**Guest**
Monitoring	•	•	•
Live streaming	•	•	•
Access notifications panel	•	•	•
View generated alarms ('Alarms' tab)	•	•	
Managing actuators	•	•	
Create/delete alarm rules	•	•	
Create/delete periodic tasks	•	•	
Send configurations to sensor node	•		

A. Sensor Nodes

1) Hardware: Fig. 3a describes the block diagram of the sensor node. It has an ATmega328p MCU [14] responsible for driving various conditioning circuits for each attached sensor/probe. The current maximum number of sensors/probes the MCU is capable of driving is limited to 10, this limitation exists due to modest number of digital pins and low dynamic RAM of the MCU. This limitation could be overcome by using a more advanced MCU if needed.

2) Software: Fig. 4 is a simplified flowchart of the deployed software on the sensor node. Commands are sent to the node via ZigBee through the serial port of the MCU. The MCU then parses these commands and actuate upon them. If an unknown command was sent, the MCU will ignore it and continue with the normal flow of the program. An agreed set of commands was designed to drive the sensor node, this set contains commands for setting/changing the sampling duration of any measured parameter, others for driving the various actuators connected to the sensor node, and the rest for over-the-air calibration of the sensors/probes attached to the node. Calculation of the sampling duration is done by starting the MCU timer at the start of the program, the timer elapsed time is then compared versus the sampling durations of all the parameters (in succession), this process is non-blocking (i.e commands sent during this process will be managed and satisfied by the MCU), if the sampling duration of a certain parameter is fulfilled, the MCU will fetch the sensor reading of that parameter and send the data to its gateway (via ZigBee). The process of fetching sensor data is blocking, but if a command was sent during this time, it will be saved in the MCU serial buffer and will be managed by the MCU once the blocking process is accomplished.

B. Gateway

Fig. 3b portrays the block diagram of the gateway. The driving hardware is a single board computer that has Ethernet/Wi-Fi capabilities. The gateway is designed as a bridge between sensor nodes and the server. The purpose of adding gateways is to extend the range of the network beyond the ZigBee limitations, thus enabling wide range installation, and more organized data flow (i.e server can process received and send data more efficiently).

C. Server

Fig. 5 illustrates the flowchart of the python script deployed on the server. The script is multi-threaded for achieving high performance. The basic number of threads are (4 + (n+1)) where n is the maximum number of parameters deployed at sensor nodes. The functions of these threads are as follow:

1) Serial port handler: reads data from and send to the serial port. After reading data from the serial port, the data is then put in a queue. The main thread will dispatch workers that handles the data present in the queue until the queue is empty.
2) Server clients handler: this thread is responsible for establishing a communication link between the server and clients connected to it via websockets. It pushes sensor data from nodes to the client and fetches all actions performed by users to be evaluated.
3) Database Workers: these threads are dispatched by the main thread, their function is to process the reading from the queue, send it to users, check if the reading satisfies an alarm rule created by Administrators/Operators (if yes, raise an alarm), and save the data to the database. The number of Database Workers are set to (n+1) as aforementioned.
4) Averaging Task: this thread becomes active every 1 minute. It fetches sensor readings from the database (24 hours interval), calculates their average, and pushes the data to users. Thus enabling users to read accurate up-to-date average values of all the measured parameters.
5) Repetitive Task Handler: this thread handles the periodic tasks feature by sending the appropriate command to its corresponding sensor node according to the task set by Administrators/Operators.

D. Web Application

The web application as shown in Fig. 6 was designed with the latest Google's design language (Material Design), and responsive design technology; Which focuses onto making the content accessible on any device with any screen resolution effortlessly.

The website is divided into 4 main section (tabs) as follows:

- Monitor: where all tiers of users can observe the readings from sensor nodes. Additionally, live streaming option is available through mainstream IP cameras. Actuators switches are available but only usable to Administrators/Operators (are grayed-out to guests).
- Alarms: generated alarms reside here, only the alarms occurred during a period of 24 hours will be shown for convenience (the rest are stored in the database).
- Alarm Rules: Administrators/Operators can view all the created alarm rules and/or periodic tasks or delete existing ones.
- Settings: Administrators can change the sampling duration of multiple parameters and send this configurations to sensor nodes.

The website is kept responsive by using the appropriate frameworks and any upon receiving data from the server, the website updates automatically without the need of refreshing the page.

The system is web-based meaning its neither device-dependent nor system-dependent, it offers three tiers of users (Administrator, Operator, and Guest) which of each has their own privileges as shown in Table I. Moreover, the proposed system boasts features like:

- Generated Alarms: The system generates alarms when certain alarm rule conditions are met. these rules include having a measured parameter (Temperature, pH, etc.) value either above the safe/unsafe maximum value or below. Administrators/Operators can add the alarm rules and delete existing ones. Each rule has a dwell interval ranges from 5 seconds to 12 hours or no dwell (immediate alarm). Also, each rule has accompanied actions which include sending an e-mail containing the alarm message and/or immediate specified actuation (turning on/off a specified actuator)
- Periodic Tasks: The system can perform periodic tasks defined by Administrators/Operators in certain time periods. Each task has a repetition interval and action activation interval (both range from 5 seconds to 24 hours), start time and actions. Periodic Tasks will be discussed thoroughly in the *System Architecture* section.
- Manual Actuation: Administrators/Operators can manually control various installed actuators (Heater, Feeder, Aerator, Lights, etc.).
- Notifications: Whenever an event occurs while the user is on session (i.e currently connected to the system), a notification will be added in the notifications panel containing a message and a time stamp. Notifications show up to all user tiers and is pushed (no need to refresh the web page).
- Configurable Sampling Durations: Administrators have access to a control panel where they can change the sampling duration for any of the measured parameters. This configurations are then pushed to the sensor node and are saved in the EEPROM of the Micro-controller so the node can resume working as required upon power loss.

V. Results

The proposed system was implemented, deployed and tested in the Integrated Circuits Lab (ICL) at the Faculty of Engineering, Ain Shams University (Cairo, Egypt). The period of testing was three months. During that period, the system was slightly tweaked for convenience, the server was hosted on the gateway rather than a solitary component. That authenticate the extent of resilience and flexibility the system can offer.

The website was tested heavily under different web browsers and is confirmed to support Chrome 35+, Firefox 31+, Safari 7+, IE 10+.

VI. Conclusion

In conclusion, having a system to monitor and control a fish farm offered a persistent, flexible, and easy-to-maintain solution. Having a flexible system broadens the domain and range of usage, as it can be easily expanded to monitor multiple farms. In addition, additional sensors that offer different monitoring criteria (i.e other than temperature, pH value and Dissolved Oxygen value) can be added to the

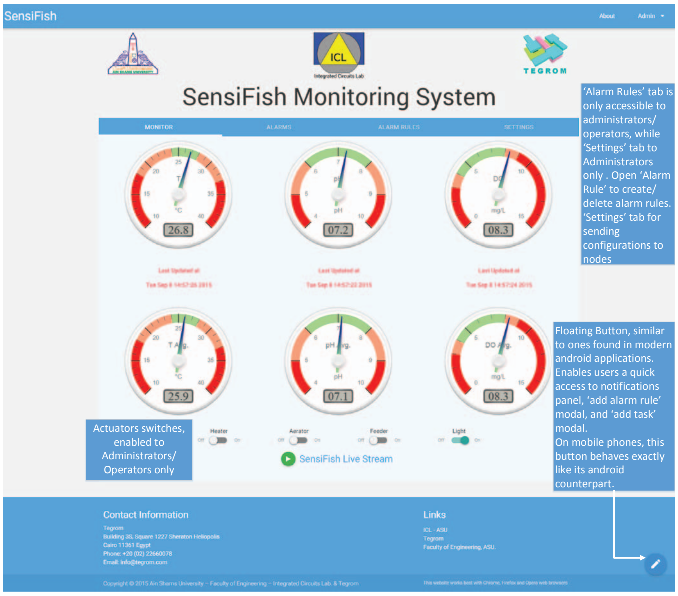


Fig. 6: Full UI capture of the website with detailed tool tips

system when needed. For example, monitoring the Ammonia (NH_3) concentration is a also a point of interest for fish farm administrators, thus by simply adding an Ammonia sensor to the system, it will enable monitoring of such value as well actuation according to it.

Persistence ensures reliability. Therefore, having a persistent 24/7 monitoring system heavily reduces human errors and increases overall reliability. Such systems are more signified in large-scale fish farms, where relying solely on a human-based solution will be much more costly and will not offer the same flexibility and reliability.

REFERENCES

[1] A. Bhatnagar and P. Devi, 'Water quality guidelines for the management of pond fish culture', INTERNATIONAL JOURNAL OF ENVIRONMENTAL SCIENCES, vol. 3, no. 6, pp. 1980-2009, 2013.

[2] M. Franson and A. Eaton, Standard Methods for the Examination of Water & Wastewater. Washington, DC: American Public Health Association, 2005.

[3] T. Wang, "Water Quality Monitoring System based on WSN", Master Thesis, ICT School of KTH, 2012.

[4] G. Xu, W. Shen and X. Wang, "Applications of Wireless Sensor Networks in Marine Environment Monitoring: A Survey", Sensors, vol. 14, no. 9, pp. 16932-16954, 2014.

[5] S. Chandanapalli, "Design and Deployment of Aqua Monitoring System Using Wireless Sensor Networks and IAR-Kick", Journal of Aquaculture Research & Development, vol. 05, no. 07, 2014.

[6] S. Chandanapalli, E. Reddy and D. Davuluri, "Efficient Design and Deployment of Aqua Monitoring Systems Using WSNs and Correlation Analysis", International Journal of Computers Communications & Control, vol. 10, no. 4, pp. 471 - 479, 2015.

[7] (2009) The Zigbee Alliance website. [Online]. Available: http://www.zigbee.org/

[8] P. Jiang, H. Xia, Z. He and Z. Wang, 'Design of a Water Environment Monitoring System Based on Wireless Sensor Networks', Sensors, vol. 9, no. 8, pp. 6411-6434, 2009.

[9] T. Wong, 'Mass fish deaths off Singapore coast spark concern', BBC, 2015.

[10] J. REILLY, 'Thousands of dead fish float to the top of volcanic Indonesian lake after being poisoned by sulphur stirred up by storms', Daily Mail, 2014.

[11] Chandanapalli SB, Sreenivasa Reddy E, Rajya Lakshmi D (2014) Design and Deployment of Aqua Monitoring System Using Wireless Sensor Networks and IAR-Kick. J Aquac Res Development 5: 283. doi:10.4172/2155- 9546.1000283

[12] Z. Rasin and M. Abdullah, 'Water Quality Monitoring System Using Zigbee Based Wireless Sensor Network', International Journal of Engineering & Technology, vol. 09, no. 10, pp. 14-18, 2009.

[13] J. Huan, X. Liu, H. Li, H. Wang and X. Zhu, 'A Monitoring and Control System for Aquaculture via Wireless Network and Android Platform', Sensors & Transducers, vol. 169, no. 4, pp. 250-256, 2014.

[14] Atmel Corporation, '8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash', ATmega328p datasheet, Aug. 2008[Revised Oct. 2009].