OYSTERS' GAPE MEASUREMENT SYSTEM

A. Ali [1], K. Ali [2], A. Ukpebor[2], M. Hasan[2], J. Addy[2] and A. Abu-El Humos[2*]

[1]*Rooster Teeth Inc*

[2]*Department of Computer Science, Jackson State University, Jackson, MS 39217, USA*

[*]*Corresponding Author Email: ali.a.humos@jsums.edu*

Full/Regular Research Papers/ISOT

ABSTRACT:

The Mississippi Based RESTORE Act Center of Excellence (MBRACE) is a consortium of four research universities: Jackson State University (JSU), Mississippi State University (MSU), the University of Mississippi (UM), and the University of Southern Mississippi (USM), where USM takes the lead in the research. MBRACE sponsored four projects in studying the importance of the environmental condition of oysters in the Mississippi Gulf Oyster Reefs with each university being assigned a specific project. JSU's team was tasked with the responsibility of developing biosensors for the measurement of Bivalve Valve Movement [15]. In this work, the hardware and software used in developing a bivalve gape measurement system have been explored. This system employs the Hall-effect phenomenon to accurately measure and report the gape of a bivalve. This system has been designed to operate in the field as well as in a laboratory environment.

Keywords: *Oysters' Gape Measurement, Hall-Effect Sensor, Arduino Nano*

[1] SYSTEM ARCHITECTURE

In this work, the hardware and software used in developing a bivalve gape measurement system is explored. This system employs the Hall-effect phenomenon to accurately measure and report the gape of a bivalve [3]. The system uses a Hall-effect sensor (Symmetry Electronics' H2425) and a small magnet in which both are glued to the exterior of the shells of a bivalve. Consequently, the Hall-effect sensor reports the distance of the magnet, hence the gape opening, to a microcontroller that records and transmits the data to a ground station. This system has been designed to operate in the field as well as in a laboratory environment. In a laboratory setting, this system uses Wi-Fi or Bluetooth to transmit its data. In the field, however, and due to the lack of availability of power, two setups have been established. The first setup does not transmit data; instead, it saves the data in an SD card within the system's enclosure. The second field design transmits data through a gateway to the cloud. The system's architecture comprises of three units: one particular Microcontroller Unit, an Arduino Nano - ATmega328P, another Microcontroller Unit, the Wemos - ESP8266, and six Hall-effect sensors, HAL 2425. The block diagram of this system is shown below in **[Figure-1]**. Each sensor system has six (6) analog inputs of the ATmega328P which are connected to six (6) Hall-effect sensors. The Hall effect sensor is placed on one side of the shell of the oyster and a magnet on the other side. The system is configured in such a way that it reads six sensors and transfers the data to a repository server and then sleeps for 90ms. Next, the system wakes up, reads the sensors and sends the data to the server and sleeps for another 90ms. This cycle continues until the battery voltage drops to 10V. The data collected by the sensor system is transmitted through a network gateway to the repository server, Mbrace.xyz. The gateway is responsible for transmitting sensor data from the sensor system through a local transit network to the remote base station that provides WAN connectivity and subsequently transmits to web-server for data logging . Lastly, data visualization is the final stage of the whole process. The data collected by the server is accessed by the users and displayed through a user interface such as mobile apps and desktop display. The full architecture is depicted in **[Figure-1]**.

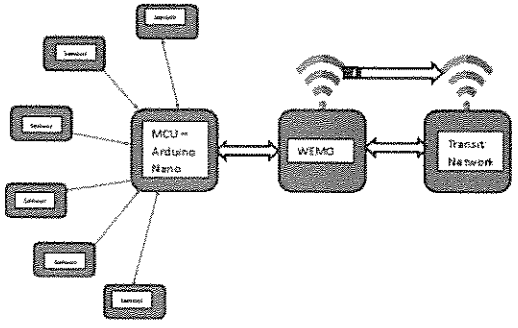

Figure: 1. Block Diagram of Sensor System

The Arduino Nano is the central engine of the sensor system. The Arduino is a microcontroller based on ATmega328p technology and is widely used in robotics embedded systems, and lightings and electronic projects where automation is an essential part of the system. It is the most accessible way of automating electronic projects, making it possible to control devices [5, 6, 7]. Another microcontroller used in this project is the ESP8266 Wi-Fi Module designed by the Chinese company, Espressif Systems [8]. It is an extremely affordable and versatile wireless System-On-Chip (SoC) that can perform the task of a wireless communication adapter for the sensor unit

978-1-7281-5584-5/19/$31.00 ©2019 IEEE
DOI 10.1109/CSCI49370.2019.00234

[11, 12]. ESP8266 is a complete Wi-Fi sensor that incorporates a 32-bit processor, some RAM, and depending on the vendor, between 512KB and 4MB of flash memory. This allows the chip to either function as a wireless adapter that can extend other systems with Wi-Fi functionality, or as a stand-alone unit that can autonomously execute elementary applications [7, 8, 15]. Lastly, this project uses the Hall-effect sensors in which the sensors and magnets are attached directly to either valve of the oysters to capture raw gape data from the animal. The Hall-effect is an ideal sensing technology. The Hall element is constructed from a thin sheet of conductive material with output connections perpendicular to the direction of current flow [1, 4]. When subjected to a magnetic field, it responds with an output voltage proportional to the magnetic field strength [1, 4]. The voltage output is very small (μV measurements in particular) and requires additional electronics to achieve useful voltage levels. When the Hall element is combined with the associated electronics, it forms a Hall-effect sensor [1, 4].

[1.1] Energy Budget and Power Consumption

In this study, we adopted several strategies to maximize the lifetime of the battery thus optimizing power consumption. One of such techniques was to disengage the LED on the microcontroller (ATmega328P) resulting in more than 50% power savings. Another method we adopted to optimize the battery power was to utilize a low power library code which causes the system to sleep between readings. This forces the sensors to power down between readings. Again, a more efficient SD card (one that consumes less power) was used in this project. The code was tweaked to make possible minimal writing to the SD card. Lastly, the supply voltage terminal of the HAL 2425 was connected to the Nano (D1 - D6) instead of the usual direct connection to the supply voltage (Vin). With this approach, the power supply to the sensors can be shut down whenever the microcontroller (Arduino Nano) sleeps. At the end of this project, a 5000mAh battery was able to last for 29 days, though the goal was to last for between 2 to 4 months on a single charge.

[1.2] Sensor Load Calculation

The Battery lasted for 29 days with a cut off voltage of 10V.

To calculate the hours DC battery will last:

Battery life in hours = rating in mAH/load in Amps ------------- (1)

Convert 29 days to hours:

29*24 = 696 hours

Battery rating = 5000mAH

Substituting these values into equation 1 above, we have:

696H = (5000*10-3) AH/I

I = (5000*10-3AH)/696H = 0.007183 A

I = 7.183mA

Consequently, the sensor load is 7.183mA.

[2] SOFTWARE COMPONENTS

[2.1] Sensor Calibration and Linearization Steps

Sensors often exhibit nonlinear transfer characteristics hence require calibration and linearization. The choice of an adequate linearization method is critical for the overall performance of the sensor system [2]. Sensors are used to convert physical quantities of interest into electrical signals. Many sense elements are inherently nonlinear. In essence, their outputs are not linearly proportional to the physical quantity they are measuring. As the physical quantity of interest changes, the output changes nonlinearly [12]. The calibration and linearization of the Hall-effect sensor define the accurate readings and actual gape distance. The goal of calibration is to minimize any measurement uncertainty by ensuring the accuracy of test equipment. Calibration quantifies and controls errors or uncertainties within measurement processes to an acceptable level [13]. The sensor provides an improved linearization output by incorporating a flexible compensation function with 16 programmable calibration points. The set-point linearization in general allows to linearize a given output characteristic by applying the inverse compensation curve [1]. In this work, Micronas HAL/HAR 24xy Programming Environment was used to calibrate and linearize the HALL 2425. To calibrate the sensors custom settings were set with magnetic range that is fitting for sensor receptiveness. A magnet was placed on the proper polarity side of the calibration machine. After that, the Arduino IDE was initiated with appropriate calibration code to perform the calibrations steps with two-point calibration method. For the linearization, in the Micronas HAL/HAR 24xy Programming Environment, the measurement file was loaded with extrapolation which results in writing set points and providing linearized values.

[2.2] Low Power Code

The purpose of this code is to make sure that the sensors use the minimum power consumption so that data can be received for a maximum timeframe. There are four (4) data collection speeds that support eight (8) sensors. These speeds are 1 = 10 HZ; 2 = 1 HZ; 3 = 1 min and 4 = 1 hour that takes reading overtime. The code runs within a loop which reads certain bytes of data with the given data collection speeds and adds these to data array. When the array is full, it writes the data on a file on the SD card. Otherwise it goes back to the loop to collect the required bytes of data.

[2.3] The Node Code

The node code comes in two flavors, one is the node code with temperature and the other one is the node code without temperature. The node code with temperature sends six bytes of data for the sensors and one byte of data for the temperature upon request from the Wemos. After reading data from all the sensors, including temperature, it returns the data in array format as shown in **[Figure-2]**. For the node code without temperature, the software follows the same structure and procedure as previously mentioned in the node code with temperature except that, it does not contain a temperature sensor. In the case of the code without temperature, it receives request and sends back 6 bytes to the Wemos.

Pseudocode of Node code with temperature

```
Setup:
Initialize WiFi Settings
Initialize I2C and Serial
Communications
Generate an Interrupt every
100 ms
Loop:
if data-array full
Send Data to Server
else
```

Figure: 2. Block Diagram of Sensor System

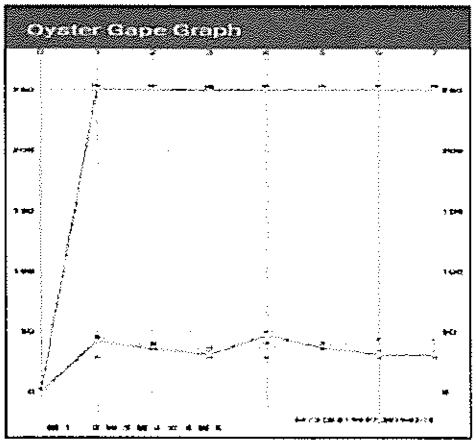

Figure: 3. Visualization of Data

[2.4] The Wemos Sequential Code

In this code, Wemos requests data to the Nano and receives six bytes of data in return and creates an array. The interrupt is set for 100 milliseconds. With every interrupt, a new request is sent, and new bytes of data were received. With this procedure, the array becomes increasingly substantial and gets assembled amongst each other. When the array size reaches maximum capacity, which is 78 bytes, it sends data to the MBRACE.XYZ server. The sending of request to the Nano continues until the array is full.

[2.5] Mobile App for Visualization

To facilitate the visualization of the data received from the sensors, a user-friendly application was developed with the Swift environment that supports the iOS devices. To view the data in the application, the host needs to be connected. With the selection of the desired date and sensor number in the application, data received from the selected sensor will be visualized on the application screen as shown in **[Figure-3]** above.

[3] RESULTS

[3.1] Gape Data Visualization

A sensor system was deployed in the field at Ocean Spring on August 10, 2018. The oysters in the field generated continuous data that was collected and transmitted to a repository server. Two sets of MATLAB programs were developed to retrieve data from Mbrace.xyz. The first program was designed in such a way that it collects live gape data while the second code retrieves old gape data. In this paper, we present a set of graphs for the sensor system. The duration of data readings is 24 hours - between the periods of 7:00 PM to 6:59 PM.

[3.2] Sensor Graph

The graph shows continuous gaping activities for 6 oysters for duration of 24 hours. The variations in colors in the graphs represent each oyster. However, one of the 6 oysters, indicated by the purple line, exhibited a different behavior from the other 5 animals where continuous recording of the valve movements were observed. The flat lining indicated at the beginning of the day was due to inactive behavior of the oyster. The oyster with purple lining in **[Figure-4]** demonstrated unique behavior from the other five oysters. It may have been due to response to the environment with unusual activities during the 24 hours period under review and perhaps there was a short circuit on the HAL 2425 sensor.

The portions of the graph that display long periods where the line is flat (i.e. the shell is open), interspersed with sporadic spikes indicates shell closure of a short duration.

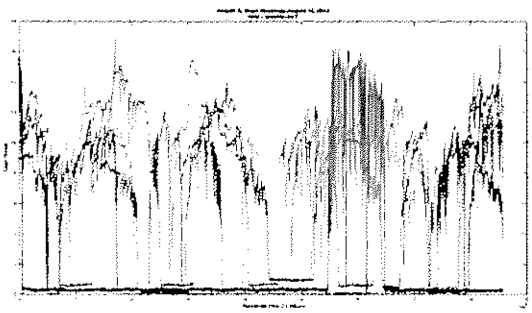

Figure: 4. Sensor gape reading, August 12, 2018

[4] CONCLUSION AND FUTURE WORK

In this research, we make use of the advances in electronic and computer technology utilizing microcontroller and Hall effect sensor to build a resilient, low-power, low-cost and multifunctional sensor systems that can measure and record the gape of bivalves (oyster) both in the field and in the laboratory. One of the goals of this research which was to build and develop an efficient power system using low power libraries that could sustain the sensor system for a long period of time was realized.

The sensor system is battery driven with limited energy sources and large power consumption. Hence, efficient energy management is a critical design objective both in hardware and software design. One of the greatest challenges of this research was the energy consumption of the system. Therefore, reducing the overall energy consumption of the system was necessary for optimal performance while satisfying the QoS (Quality of Service) or performance constraints.

ACKNOWLEDGEMENT

This project was paid for with federal funding from the Department of the Treasury under the Resources and Ecosystems Sustainability, Tourist Opportunities, and Revived Economies of the Gulf Coast States Act of 2012 (RESTORE Act). The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the Department of the Treasury.

REFERENCES

[1] TDK Micronas, "HAL/HAR 24xy User Manual", 2016

[2] Antonio J. Lopez-Martin, Alfonso Carlosena, "Sensor signal linearization techniques: A comparative analysis", https://ieeexplore.ieee.org/document/6519013, 2013.

[3] Symmetry Electronics, "Hall Effect Sensors", https://www.semiconductorstore.com/blog/2017/Tech-101-From-Symmetry-Electronics-Hall-Switches/2791, 2017

[4] Electronics Tutorial, "Hall Effect Sensors", "https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html", 2018

[5] Electronics Source Co.,Ltd, "Arduino Nano 3.0" https://www.es.co.th/Schemetic/PDF/ARMB-0022.PDF, 2017

[6] Mouser Electronics, "Arduino Nano Specification Sheet", https://eu.mouser.com/catalog/specsheets/Arduino-ArduinoBoardNano.pdf, 2016

[7] Robo Romania, "How to use Arduino Nano Programming Environment", http://roboromania.ro/datasheet/Arduino-Nano-roboromania.pdf, 2017

[8] Espressif Systems, "Specification of ESP8266EX", https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf, 2018

[9] Geekstips, "ESP8266 Arduino Tutorial – Wifi Module Complete Review" https://www.geekstips.com/esp8266-arduino-tutorial-iot-code-example/, 2018.

[10] Random Nerd Tutorials, "Getting Started with ESP8266 WiFi Transceiver", https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/, 2015

[11] Electronics Foru, "10 Awesome ESP8266 Projects for Electronics DIYers", https://electronicsforu.com/electronics-projects/10-esp8266-projects, 2017

[12] EDN Network, "Linearization of sense element outputs using sensor signal conditioners", https://www.edn.com/design/analog/4442094/Linearization-of-sense-element-outputs-using-sensor-signal-conditioners, 2016.

[13] Tempcon Instrumentation, "What is calibration and why is it so important", https://www.tempcon.co.uk/2018/02/26/what_is_calibration/, 2016

[14] Mbrace, "Mississippi Based RESTORE Act Center of Excellence ", https://mbrace.usm.edu/, 2017

[15] ESP8266, "The Internet of Things with ESP8266", http://esp8266.net/, 2016