Interfacing Arduino with IoT: A Water Pressure Monitoring System

V. Balbarino, J. Basan, R. Cerna, S. Larena, R. Tabucanon
Bachelor of Science in Mechanical Engineering
Visayas State University
Baybay City, Philippines
vicbalbarino312@gmail.com
jojulestelin@gmail.com
arnerisbby@gmail.com
seanclejlarena@gmail.com
romulotabucanon122802@gmail.com

***Abstract*— Continuous monitoring of water pressure is essential to attain maximum system performance in different applications, and with the rise of Internet of Things (IoT) technology, an Arduino-based system that gathers, processes, and transmits data from a pressure sensor to a cloud-based platform presents a real-time, cost-friendly alternative to conventional monitoring systems, enabling remote access, quick issue identification, and integration with other IoT devices for increased efficiency and reliability in water supply systems. The study aims to manufacture a portable water pressure monitoring system with IoT capabilities, which will involve designing a system with a pressure transducer sensor, incorporating Arduino Uno R3 and ESP8266 Wi-Fi Module for data logging and analysis, calibrating the system, conducting system validation, and displaying the finished monitoring device. For constructing the system, the researchers carefully considered the hardware and software development process. The researchers integrated IoT into the system, including data transmission from the system to the database, with the help of a database and an MIT application creator. By comparing the measurement from the pressure monitoring system to an analog pressure gauge, the researchers manually calibrated the water pressure monitoring system within time intervals and recorded the readings. The correlation between the information acquired by the pressure monitoring system and the analog pressure gauge was analyzed and modeled using the T-test, Kolmogorov-Smirnov Test of Normality, and Levene's Test of Homogeneity of Variance. The values of an analog pressure gauge and a pressure transducer were compared using the statistical tools mentioned and showed that there was no discernible difference between them.**

Keywords*— *Pressure, Internet of Things (IoT), Arduino UNO R3, Database, T-test, Kolmogorov, Levene's Test, Calibration, ESP8266 Wi-Fi Module

I. Introduction

A. Nature and Importance of the Study

Water pressure is critical in various applications, such as water supply systems, irrigation systems, and industrial processes. In these applications, water pressure must be maintained within a specific range for optimal system performance. One of the significant factors influencing leakage is the pressure in a water distribution system [1] [2]. Water supply systems with low pressure can lead to insufficient water flow, while high water pressure can cause damage to the pipes and other components of the system. Therefore, it is essential to monitor water pressure continuously to ensure the proper functioning and efficiency of the system.

Traditionally, water pressure monitoring systems have been implemented using wired sensors and meters that require manual reading and recording of data. These systems are often expensive, time-consuming, and require frequent maintenance. With the advent of IoT technology, it is now possible to implement water pressure monitoring systems that can be remotely monitored and controlled in real-time.

Arduino is an open-source microcontroller that can easily be programmed, erased, and reprogrammed at any instant in time [3]. It is widely used in IoT-based projects due to its low cost, ease of use, and availability of sensors and modules. In the proposed system, an Arduino board collects and processes data from a pressure sensor that measures water pressure. The microcontroller then communicates with the ESP8266 Wi-Fi module, where the data is transmitted to a cloud-based platform, where its users can store and view it.

The IoT-based Arduino water pressure monitoring system has several advantages and offers an attractive alternative to traditional monitoring systems. It is cost-effective, easy to install, and provides real-time data that can quickly identify issues. The system can be accessed remotely using a smartphone or a web interface, allowing users to monitor the water pressure in real-time and receive alerts in case of abnormalities. Moreover, such systems can be integrated with other IoT devices, such as smart valves and pumps, to create a complete IoT-based water supply system.

The proposed IoT-based Arduino water pressure monitoring system can improve the efficiency and reliability of water supply systems by providing real-time monitoring and control. It can contribute to developing sustainable water management systems and pave the way for further research in IoT-based water supply systems.

II. Objectives

A. Objectives

Generally, this study aims to develop a portable water pressure monitoring system with the feature of Internet of Things (IoT).

Specifically, this study aims to:

1. Fabricate a water pressure monitoring system that employs a pressure transducer for measuring water pressure in pipes;
2. Interface Arduino with IoT capabilities for data logging and analysis; and
3. Test the accuracy and reliability of the monitoring system by calibration and statistical analysis.

B. *Scope and Limitation*

This study is focused only on the design and fabrication of an IoT-based Arduino Water Pressure Monitoring System, which involves building a prototype system that consists of an Arduino board, a pressure transducer, and a Wi-Fi module and developing a server and an app that can display real-time data. This study is limited to monitoring water pressure on small-scale water distribution networks and will not address other factors affecting water quality and supply.

III. METHODOLOGY

A. *Materials*

- 1 pc. Plexiglass (2x2)
- 1 pc. Pressure Gauge
- 1 pc. UNO R3 board
- 1 pc. Pressure Transducer Transmitter Sensor
- 1 pc. ESP8266 ESP-12E WiFi development board
- 1 pc. 20x4 LCD Display I2C
- 1 pc. PCB Board
- 2 pcs. GI Bushing RED
- 2 pcs. GI Coupling RED
- 2 pc. GI Nipple S40
- 1 m. Stranded Wire
- 1 pc. Switch
- 1 pc. Eveready 9V Battery

B. *Circuit Diagram*

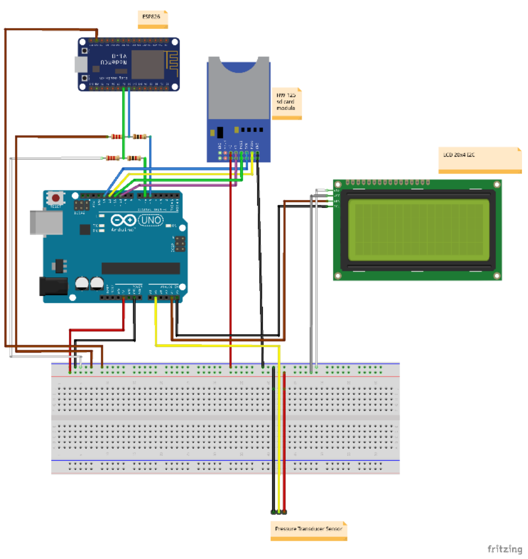

Figure 1. Prototype Breadboard Wiring Diagram

C. *Hardware System Configuration*

Referring to the prototype breadboard wiring diagram, analog pin A1 is used to connect the analog wire of the pressure transducer. Consequently, the ground and voltage pins of the pressure transducer is connected to the GND pin and 5V, respectively. With the regards to the HW-125 SD card module; the GND pin to the common ground wires. Master In, Slave Out (MISO) to the digital pin 12, Serial Clock (SCK) to the digital pin 13, Master Out, Slave In (MOSI) to the digital PWM pin 11, Chip Select (CS) to the digital PWM pin 10. Moving on the LCD 20x4 I2C, VCC, GND, Serial Data (SDA), and Serial Clock (SCL) is connected to the common 5 volts wires, common ground wires, analog pin A4, and analog pin A5, respectively. Shifting the focus to the ESP8266, since the ESP8266 module doesn't have a built-in voltage regulator, voltage level shifters were created. 1K and 2K-ohms resistors were used to level shift the 5V to a safe and passable value of 3.3V. This is to avoid high voltage values that are detrimental to the ESP8266 if allowed.

D. *Software System*

1) *System Microcontroller*

The system uses an Arduino Uno R3 microcontroller programmed in C++ style to process data from the input device and display it on an LCD output. The code includes parts for configuring libraries and defining pins, initializing modules, converting analog signals to digital, converting digital pressure to pressure measurements, storing data in a software document, and data which is connected to the ESP8266 WiFi Module for communication and database connectivity

The ESP8266 WiFi Module is an important component that connects the system to the database and allows data exchange and communication with other devices or cloud services The code contains libraries for WiFi, JSON data processing, Firebase database, and software of serial communication.

The code defines constants for WiFi credentials, API key, and database configuration. It contains commands for configuring Firebase, managing and validating data, and configuring software-based serial ports. The "Setup" function initiates serial communication with WiFi modules. There is also a provision to create a static JSON document for communication with the WiFi module, with a counter to delete the document after a certain amount of data has been transmitted.

2) *Application Development and Integration with Internet of Things*

In the context of the Internet of Things (IoT), researchers integrated a water pressure monitoring system with IoT capabilities using an application. The Arduino Uno R3 microcontroller receives the sensor data and converts it into a pressure value, which is displayed on the LCD screen. Through serial communication, the Arduino transmits the pressure value to the WiFi module of the ESP8266. The module then sends the data to the Firebase Realtime Database and the ThingSpeak channel created by the researchers. This allows the researcher's application, developed using MIT App Inventor 2, to display the stored data. Users can monitor pressure in real time, either remotely or instantly. The researchers also implemented notification alerts in the process to notify users when pressure data exceeds a certain threshold.

3) *Connecting the Application*

To connect the Water Pressure Monitoring Application, a Firebase database needs be set-up first. This can be done by setting up the Firebase console where the Firebase Realtime Database is enabled to obtain necessary credentials like the Firebase API key for authentication. In the MIT App inventor, the app was created and connected to the Firebase Database by adding the Firebase component to the application.

In the app, the Firebase component blocks are used to create a connection within these two. Data is retrieved from the Firebase database using the "Get Value" or "Get Tag List" blocks, specifying the location in the database where the data will be called. "Set Value" or "Store Value" blocks are used for storing and updating data in the firebase, and blocks like "Data Changes" or "Tag List Changed" handle real-time updates. With this, the Firebase database offers real-time data. Once the data is retrieved and updated, it will be then integrated into the app's logic, user interface, or other visual blocks-based coding interface in the MIT App Inventor.

E. Calibration

This calibration process involves comparing the sensor's readings from the pressure transducer to the analog pressure gauge using a T-Shaped connection pipe. Both pressure measuring device were attached to the ends (refer to Figure 2).

Figure 2. Analog Pressure Gauge and Transducer Set-up

Pressures were recorded from both devices simultaneously where readings were taken with an interval of 30-seconds for every 2 mins from the device while readings from the analog pressure gauge were taken every end of 2 minutes. This resulted to 20 mean pressure readings for both pressure transducer and the analog pressure gauge.

To analyze and compare the calibration data, the researchers employed statistical tests; T-Test, Kolmogorov-Smirnov Test of Normality, and Levene's Test of Homogeneity of Variance. These tests assessed the relationship between the acquired data and used the following hypothesis:

Null Hypothesis
Ho:
1.The samples drawn are normally distributed.
2. There is no significant difference between the analog pressure gauge and the pressure transducer variance values.
3. There is no significant difference between the value of the analog pressure gauge and the pressure transducer.

Alternative Hypothesis
Ha:
1. The samples drawn are not normally distributed.
2. There is a significant difference between the analog pressure gauge and the pressure transducer variance values.
3. There is a significant difference between the value of the analog pressure gauge and the pressure transducer.

IV. Results and Discussions

Following the proposed hardware design and system, the researchers have mainly integrated the suitable hardware components: the Arduino microcontroller, pressure transducers, and wireless communication modules.

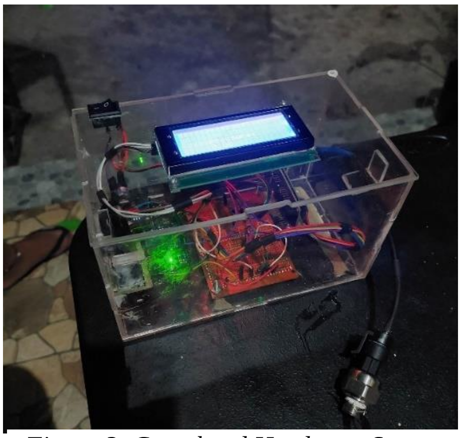

Figure 3. Completed Hardware System

A. System Validation

To calibrate the water pressure monitoring system, the researchers took a manual approach by comparing it to an analog pressure gauge. The researchers positioned the two devices side by side to directly observe the differences in their readings as shown in Figure 2. By this method, we were able to align the readings of our device with those of the analog pressure gauge, and were able to achieve the following results.

Table 1. Significant difference between the Analog Pressure Gauge and the Pressure Transducer

Statistical Tool Used Two-Tailed T-Test for 2 Independent Samples	T value computed	α	df	p-value obtained	Interpretation	Decision
Trial 1 (Analog and Pressure Transducer)	-0.07906	0.05	19	0.793876	There is no significant difference between the value of the analog pressure gauge and the pressure transducer.	Fail to reject the null hypothesis.
Trial 2 (Analog and Pressure Transducer)	-0.09124	0.05	19	0.927784	There is no significant difference between the value of the analog pressure gauge and the pressure transducer.	Fail to reject the null hypothesis
Trial 3 (Analog and Pressure Transducer)	-0.05702	0.05	19	0.954831	There is no significant difference between the value of the analog pressure gauge and the pressure transducer.	Fail to reject the null hypothesis

Table 1 above shows the results for the conducted T-Test for two independent variables to compare the manual readings obtained from the analog pressure gauge and the pressure transducer in the device. The three trials obtained computed t-values that were lower than the critical values at 95% confidence interval and 19 degrees of freedom. These results lead to the acceptance of the null hypothesis, which indicates that there is no significant difference between the values obtained from both pressure measuring devices.

Table 2. Kolmogorov-Smirnov Test of Normality for Analog Pressure Gauge and Pressure Transducer

Statistical Tool Used: Kolmogorov-Smirnov Test of Normality	K-S test statistic value	Skewness	n	p-value obtained	Interpretation	Decision
Trial No. 1 (Analog)	0.24838	0.089524	20	0.14237	The data does not differ significantly from that which is normally distributed.	The samples are normally distributed.
Trial No. 1 (Digital)	0.24021	0.043704	20	0.16839	The data does not differ significantly from that which is normally distributed.	The samples are normally distributed.
Trial No. 2 (Analog)	0 .25013	0.0200311	20	0.13724	The data does not differ significantly from that which is normally distributed.	The samples are normally distributed.
Trial No. 2 (Digital)	0 .2396	0.110021	20	0.17046	The data does not differ significantly from that which is normally distributed.	The samples are normally distributed.
Trial No. 3 (Analog)	0.24895	0.006742	20	0.1407	The data does not differ significantly from that which is normally distributed.	The samples are normally distributed.
Trial No. 3 (Digital)	0.23648.	0.040424	20	0.18145	The data does not differ significantly from that which is normally distributed.	The samples are normally distributed.

Table 2 above shows the results of the Kolmogorov-Smirnov Test of Normality, which examines the normal distribution of the analog pressure gauge and pressure transducer readings. The biggest absolute difference (D) between the two CDFs is employed as the test statistic. The results obtained from the six samples over the three trials show that the data collected does not stray significantly from a normal distribution. As a result, we can conclude that the samples are normally distributed.

Statistical Tool Used: Levene's Test	f-ratio value	α	df within treatments	p-value obtained	Interpretation	Decision
Group No. 1	0.00021	0.05	39	.988625	There is no significant difference between the variances between the analog pressure gauge and the pressure transducer.	The required homogeneity is met.
Group No. 2	0.00115	0.05	39	0.97317	There is no significant difference between the variances between the analog pressure gauge and the pressure transducer.	The required homogeneity is met.
Group No. 3	0.00524	0.05	39	0.942697	There is no significant difference between the variances between the analog pressure gauge and the pressure transducer.	The required homogeneity is met.

Table 3 above presents the results of the Levene's Test which was conducted to compare the variances of both pressure measuring devices. The results showed that the computed f-values in all three trials were smaller than the critical values at 95% confidence interval. This leads to the acceptance of the null hypothesis and indicates that the variances between the analog pressure gauge and the water pressure monitoring device are not significantly different. Thus, the required homogeneity is met.

B. Software System Evaluation

A software system evaluation was also done to make sure that the application receives the data from the device. Figure 4 below shows the working Realtime Database.

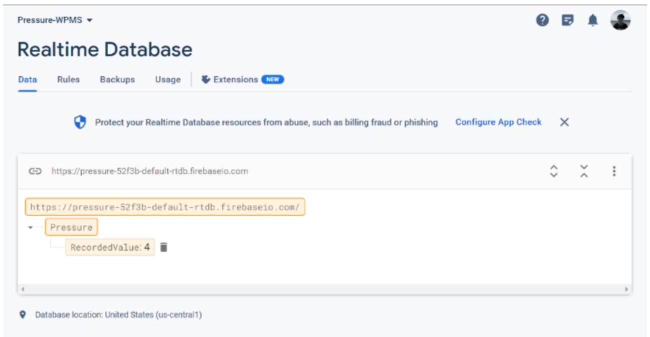

Figure 4. Real-time Database

The ESP8266 writes the data communicated by the Arduino to the ThingSpeaks channel created by the researchers. The transmitted data is visualized into a line graph by utilizing the website's automatic tabulation. The x-axis is the date and time, while the y-axis is the pressure data.

As shown in Figures 4 and 5, the application graphs the changes in pressure as the Real-time Database receives data.

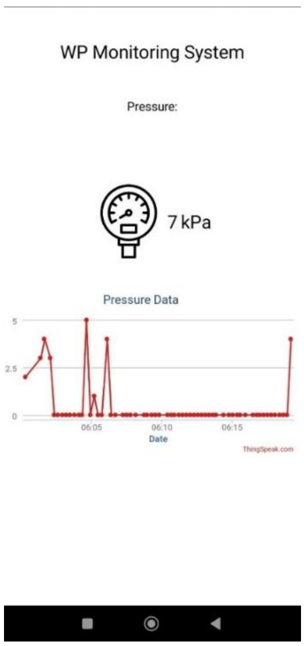

Figure 5. Application Interface

V. Conclusions and Recommendations

This study intended to develop a water pressure monitoring device that integrates the Internet of Things capabilities to acquire real-time data. By programming the device to capture and transmit the system's readings to a dedicated server and app, data logging and analysis became more efficient and provided a more comprehensive analysis.

The system's ability to obtain real-time data provides significant advantages for monitoring water pressures in various applications. The advantages include immediate detection of pressure change or fluctuations, leading to timely response to anomalies or issues. With the help of the internet and the app, it provides a more user-friendly interface for data logging, visualization, and analysis that enhances the accessibility and usability of the monitoring system.

To test the accuracy and reliability of the monitoring system the researcher used calibration with the help of statistical tool namely T-Test, Kolmogorov-Smirnov Test of Normality, and Levene's Test of Homogeneity of Variance. It was observed that there is no significant difference between the analog pressure gauge and the pressure transducers. It was also observed that there is a significant relationship between the analog pressure gauge and the pressure transducers.

From the hardware and software evaluation that has been conducted, it is concluded that the water pressure monitoring system functioned with accuracy and can be utilized to perform real-time water pressure monitoring. The hardware system shows complete circuit build-up and the software system shows no malfunction. Real-time pressure readings are received by the Realtime Database and the application.

The researchers suggest substituting the Arduino UNO R3 with the Arduino Mega 2560 Rev3, which offers 256 KB compared to the UNO which has only 32 KB. This substitution allows more programs, libraries, and functions in the code. Additionally, the researchers recommend incorporating more components such as: RTC Module for precise time tracking, turbidity sensor, pH sensor, and temperature sensor for measuring water quality. Use of other components for measuring water quantity could also be integrated. Furthermore, the researchers also recommend the use of a more accurate pressure measuring device aside from the analog pressure gauge to further test the reliability of the system.

References

[1] T. J. a. L. A., " “Progress in Practical Prediction of Pressure/Leakage, Pressure/Burst Frequency and Pressure/Consumption Relationships”," Proceedings of IWA Special Conference 'Leakage 2005' , Halifax, Canada, September 2005, 2005.

[2] C. B., "“Experiences in DMA redesign at the Water Board of Lemesos”," Proceedings of IWA Special Conference 'Leakage 2005' , Halifax, Canada, September 2005, 2005.

[3] M. (. Banzi, Getting Started with Arduino, CA: USA: Make: Books; 1st edition, 2008.