

Design and Implementation of IoT-Based Water Pipe Pressure Monitoring Instrument

Hollanda Arief Kusuma^{a*}, Rady Purbakawaca^b, Irwan Rudy Pamungkas^c, Luthfy Nizarul Fikry^c, Sonny Seftian Maulizar^c

^aDepartement of Electrical Engineering
Maritime University of Raja Ali Haji
Jl. Polteknik Senggarang
Tanjungpinang, Indonesia

^bAgricultural Engineering Sciences
IPB University
Gedung Fateta, Kampus IPB Darmaga PO BOX 220, Bogor 16002
Bogor, Indonesia

^cCV Mantis Indonesia
Jl. Mola No.15, Komplek Cimanggu Perikanan, Kel. Kedung Waringin, Kec. Tanah Sareal
Bogor, Indonesia

Abstract

The water pressure monitoring system in the PDAM pipeline networks has been successfully developed for operation and maintenance of water leaks in a real-time manner. This research aims to design a water pressure monitoring system in operational piping networks to identify anomalies as early as possible. The system is built using a microcontroller, a 1.2 MPa fluid pressure sensor and a control system equipped with a GSM wireless communication module, an Analog to Digital Converter module with 16-bit resolution, a real-time clock peripheral, an OLED display 128 × 64, and a micro SD card. The developed system was tested in a pressure range of 0.200 - 0.800 bar with 30 repetitions with a RMSE of 0.058 bar. This system has a deterministic coefficient of 0.885 against a standard manometer. The system implemented in the field successfully sends data to the server with a success rate of 96.0%. Data is displayed on a monitoring dashboard that can be accessed via a computer or smartphone.

Keywords: water pressure, pressure monitoring, PDAM, water leakage, IoT.

I. INTRODUCTION

Internet of Things (IoT) is defined as devices that can sense, collect, sometimes analyse data, and share the data through networks across the Internet. The fundamental requirement of IoT is to connect the physical world to the digital world [1]. Currently, IoT requires capability to integrate, communicate, and analyze the data through the network. IoT communication system must include its connectivity, interoperability, and integration. A development of an IoT device is dependent on its network communication to the server. Several networks used in IoT devices are Zigbee, Bluetooth Low Energy, LoRa, mobile networks (GPRS, 3G, 4G, LTE, etc.), Wireless LAN, and Mobile Adhoc Networks (MANET) [2], [3].

Arduino is an open source hardware and software platform [4]. Arduino as an integrated development environment provide service to write, edit, and convert code into instructions that uploaded into an Arduino hardware. Processing language used as the basic programming language in Arduino Integrated Development Environment (IDE) [5]. Arduino boards execute code uploaded from IDE. Atmel's AVR 8-bit

microcontrollers generally used as the main microcontroller in Arduino Board. The code from Arduino IDE is uploaded to the Arduino Board and executed by its microcontroller [6].

Arduino provides libraries from the worldwide community. Manufacturers also support sensors and libraries that foster Arduino to be successfully implemented and become familiar to the community [7]. Several implementations have used Arduino-based platform such as Urban Traffic of the vehicle via GSM [8], Smart Energy Meter [9], [10], flood alert system for parking lots [11], temperature, humidity, and soil moisture monitoring for irrigation through GSM [12], and Geohazards Monitoring and Early Warning System [13]. These studies show that the Arduino platform is reliable and easy to use for open source hardware development.

PDAM Tirta Pakuan Bogor is a local drinking water supply company in Bogor City. At one occurrence, water supplies were cut due to pipe leakage [14]. This problem could not be detected in real-time. PDAM usually identify the problems source from user complains or regular inspections. Usually, PDAM uses an analog manometer to monitor the water pressure [15]. Then, a PDAM technician inspects on the field to investigate the real condition. This approach is not an efficient way to solve such problem. Therefore, a real-time water pressure monitoring instrument need to be built hence PDAM can

* Corresponding Author.

Email: hollandakusuma@umrah.ac.id

Received: January 11, 2021 ; Revised: February 02, 2021

Accepted: April 06, 2021 ; Published: August 31, 2021

take action as soon as possible when operational issues occur.

In response, we have developed an IoT Water Pressure Monitoring System to monitor water pressure in PDAM pipe network that possess the following features:

- Designed and constructed using both open source hardware and software;
- Long-term onboard data storage on SD cards at user-defined intervals;
- Capable to measure water pressure with acceptable accuracy ($\pm 10\%$);
- Have a real-time clock;
- Able to send the data to server periodically.

II. METHOD

The main objective of this work is to obtain accurate water pressure data from a water pressure sensor. The proposed system has three layers: acquisition, transport, and application layer. The acquisition layer consists of a water pressure sensor, an ADS1115, an Arduino Mega2560, a Real-Time Clock (RTC) DS3231, an OLED display 128×64 and a micro SD card. Data received from the sensor were sent to an application layer through the transport layer using GSM module SIM900A.

A. Acquisition Layer

This prototype built was based on Atmel ATmega2560 microprocessor. ATmega2560 have 256 KB flash memory, 8 KB SRAM, and 4 KB EEPROM to run the program. An ATmega2560 has four USART. This instrument uses only two serials for serial debugging and GSM communication. Arduino Mega2560 clone, namely RobotDyn Arduino Mega2560 Pro Mini was chosen as the microprocessor board due to its compact size (38×52 mm) compared to Arduino Mega2560 (53.34×101.6 mm).

Time measurement was provided by Maxim DS3231. DS3231 uses I2C communication to send time data to Arduino Mega2560. RTC has the capability to conduct time-keeping consistently without the dependency on the Arduino power supply [16].

ADC 16-bit ADS1115 was used to convert analog value (voltage) from water pressure sensor to digital value. ADS1115 is able to perform high-resolution analogue to digital conversion, hence a more precise measurement value is acquired. Single ended conversion was employed to obtain the sensors data in order to measure the positive voltage. By using this conversion, ADS1115 acquired 15-bit resolution.



Figure 1. Water pressure sensor 1.2Mpa.

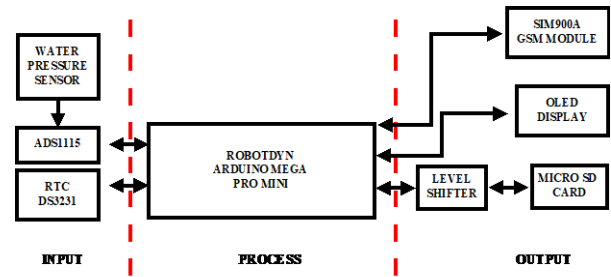


Figure 2. Components diagram of the acquisition layers.



Figure 3. SIM900A GSM module.

Water Pressure Sensor G1/4 1.2 MPa was employed to obtain the pressure value shown in Figure 3. The sensor working input voltage is at 5 V, the range of output voltage is at 0.5 V – 4.5 V, and the pressure range is at 0– 1.2 MPa with a maximum pressure of 2.4 MPa. The sensor was connected to ADS1115 analog input channel 0. To obtain pressure value, we used formulation from (1) and (2). Analog values from pressure measurement were converted to digital numbers (DNs). DN's were multiplied with 0.1875 since 1-bit is equal to 0.1875 mV and were divided by 1000 to convert the unit from mV to V. Equation (2) uses a linear regression from a minimum value (0 Pa, 0.5 V) and a maximum value (1.2 MPa, 4.5 V) with offset defined by a user. The components diagram and connection of the device shown in Figure 2.

$$V = \frac{DN \times 0.1875}{1000} \quad (1)$$

$$P = (300 \times V - 150) \times 0.01 \quad (2)$$

where:

V = Voltage (Volt)

P = Pressure (bar)

B. Transport Layer

Transport layer in this work consists of GSM module SIM900A produced by SIMcom. Figure 3 shows a module which runs with power of 5 VDC supply. The module connects with Arduino Mega2560 via serial communication (Serial1) with baudrate 9600 bps. AT commands are used in this module to send water pressure sensors data to the server [17].

C. Application Layer

Web User Interface (UI) was used to display water pressure sensor data sent from the device. This web UI were designed to be responsive so the data are easily accessed using smartphones and computers.

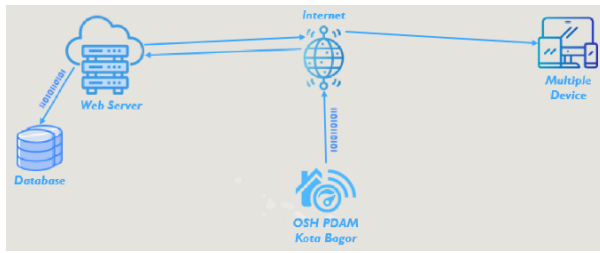


Figure 4. Working principle of the proposed web user interface.

The web also provides login to authorize users to access the data. This web UI was built using HTML5, CSS3, JavaScript, MySQL, bootstrap, Highcharts and PHP. In this work, the RESTful application interface with POST method was used to send water pressure sensor data to the cloud service. The Web UI working principle is shown in Figure 4.

D. Data Analysis

Accuracy is one of the most critical factors to consider in selecting the sensor to detect water pressure. The accuracy of the sensor is described as how close a measurement value to the actual value of water pressure. In this research, manometer analogue was used as reference.

Each singular measurement in n samples contains an observation error value (e_i) that influences its accuracy. To assess the accuracy of the measurements, the relative error of measurement (ϵ_r) was used based on the ratio of observation error and actual value, as shown in (3) [18]. Root mean square error (RMSE) in (4) was used to obtain the total error value based on [19].

$$\text{Relative error } (\epsilon_r) = \frac{\text{Absolute error}}{\text{True value}} = \frac{\delta A}{A} \quad (3)$$

$$\text{RMSE} = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n e_i^2} \quad (4)$$

Comparison between the water pressure sensor and manometer was provided by using linear curve fitting. The regression model by using coefficient of determination (r^2) then was employed to evaluate the fitting. The coefficient of determination can be interpreted as the percent reduction in the total variation in the experiment obtained by using the regression line [20], [21].

III. RESULT AND DISCUSSION

A. System Programming Design

This instrument was developed for measuring water pressure from water pressure sensor, calculating the data, generating water pressure value, displaying it on an OLED display, storing the data in the micro SD card, and sending the data to the server. This instrument is capable for measurement based on interval that are saved in micro SD card. The instrument firmware was built using Arduino IDE. Figure 7 shows the flowcharts of this system.

Arduino Mega2560, ADS1115, RTC, and OLED display are communicated through the I2C protocol using library Wire.h. Micro SD card communicates with Arduino Mega2560 using SPI protocol. Data received from the water pressure sensor are converted using ADS1115. Arduino Mega2560 UART1 connected with SIM900A. In this firmware, external library such as RTCLib for the RTC was added, SDFAT for accessing the micro SD card, Adafruit_ADS1015 for communicating with ADS1115, Adafruit_SSD1306 and Adafruit_GFX for OLED display, TimeLib, and TimeAlarm to set sampling time.

The program begins with serial communication and OLED display initialization using I2C address $0 \times 3c$. The microcontroller instructs OLED to display the text “USAID- PDAM OPEN SOURCE HARDWARE WATER PRESSURE SENSOR DEVICE 2019”. This text become an indicator to ensure that the device is on and working properly. Then microcontroller checks whether RTC DS3231 is connected or not using *RTC.begin()*. If microcontroller detects that RTC is not connected then the next command is not executed, then the OLED display text “Error! Contact CS”. If RTC initialization complete, then microcontroller obtain the time and date data from RTC. These data are displayed on OLED in three seconds. After that, the micro SD card initialization begins. If pin Select detects the absence of micro SD card, then OLED displays text “SD card Error!!!”. If micro SD is detected, then the text “SD Card OK!” is displayed on OLED.

Next step is checking **config.txt** inside the micro SD card. This text file contains ID Station, data sampling interval, and bursting data configured by a user. Sampling interval is specified in minutes and burst data is in seconds. If **config.txt** does not exist, the microcontroller instructs OLED to display text “Configuration file Error Reading”. After acquiring data from **config.txt**, ADS1115 initialization is performed followed by SIM900A initialization. Microcontroller instructs SIM900A to check the operator name using “AT+COPS?”. SIM900A then returns the current mode and the currently selected operator. “AT+CSQ” is sent to SIM900A to check the signal quality inside the SIM Card. This command returns the received signal strength indication from 0 until 30. The last setup is setting the sampling time based on interval data. This sampling setting is done using function *Alarm.repeat* provided by TimeAlarms library.

Algorithm Get Mean DN from water pressure sensor

1. Initialize count = 0, totalDN = 0;
2. **for** (count=0; count<burst; count++){
3. DN = ADS1115.readAnalogChannel(0);
4. totalDN += DN;
5. }
6. meanDN = totalDN/burst;

Figure 5. Pseudocode to obtain average DN from water pressure sensor using ADS1115.

Algorithm Convert DN pressure (bar)

1. $Volt = \text{meanDN} * 0.1875 / 1000;$
2. $Pressure = (300 * Volt - 150) * 0.01$

Figure 6. Pseudocode to convert DN to bar.

The main program begins with taking the time data from RTC. Then the microcontroller obtains pressure data in Digital Number (DN) from ADS1115 in 2^{16} resolution using **for** looping based on burst value defined by the user.

Figure 5 shows the pseudocode in obtaining the data. The total data after looping are divided with burst interval to get average DN values. Next, conversion data from digital number to voltage is carried out using (1) and converts the voltage into pressure in bar by (2). The conversion DN to pressure pseudocode is shown in Figure 6. Subsequently, water pressure data displayed in OLED.

The next command saves time, date, and water pressure data into Micro SD card as plain text with a comma (',') as the delimiter in a *.txt file format. yyyyymmdd.txt is the name of the file inside Micro SD card. This file is created automatically by getting the date data from RTC. After the saving process into micro SD card is completed, water pressure data are sent to the server. The data are sent using JSON format. Microcontroller instructs SIM900A to send data. All AT commands that are used to execute the POST method and send the data are shown in Table 1. The commands must execute in appropriate sequence. Microcontroller wait SIM900A to give response "OK" in timeout period. The microcontroller waits until the next time interval to retrieve water pressure data again. During the waiting period, microcontroller acquires date and time data from RTC and display them on OLED. The firmware in this system works is based on the flowchart shown in Figure 7.

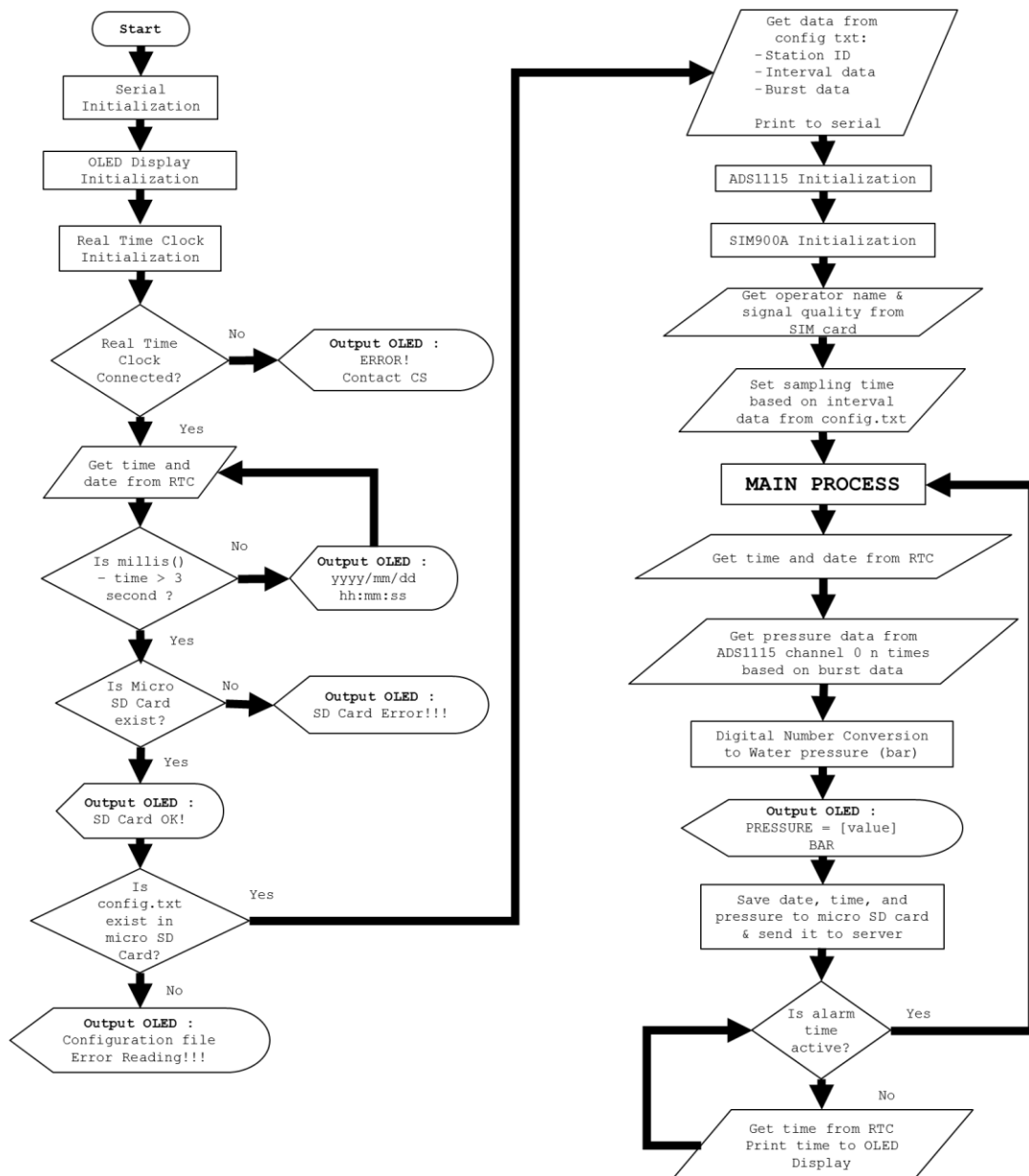


Figure 7. Flowchart of the proposed instrument.

TABLE 1. LIST AT COMMAND USED TO SEND DATA TO SERVER

AT COMMAND	DESCRIPTION
AT+CGATT=1	Attach GPRS service
AT+CIPSHUT	Deactivate GPRS PDP context
AT+SAPBR=0,1	Close bearer
AT+SAPBR=3,1,"CONTYPE","GPRS"	Set bearer connection type to GPRS
AT+SAPBR=3,1,"APN","Internet"	Set bearer APN to Internet
AT+SAPBR=1,1	Open bearer
AT+SAPBR=2,1	Query bearer to get IP address
AT+HTTPTERM	Terminate HTTP Service
AT+HTTPINIT	Initialize HTTP service
AT+HTTTPARA="CID",1	Set HTTP Bearer profile identifier value into 1
AT+HTTTPARA="URL","http://www.mantisid.id/api/product/pdam_dt_c.php"	Set HTTP client URL value
{'Data': 'yyyy-mm-dd hh:mm:ss', 'water pressure', 'burst interval', 'data interval'}	Data in JSON format
AT+HTTPDATA = json.length(), 15000	Input HTTP data with maximum time input data 15000 milliseconds
AT+HTTPACTION=1	HTTP method action set to POST method

B. Water Pressure Sensor Accuracy

Sensor accuracy is evaluated by comparing the prototype with a manometer analogue. From 30 repetition of observation, RMSE value is 0.058 bar. Relative error of this sensor is 0.09 or 9%. This value is statistically acceptable; therefore, the accuracy is satisfactory. We also evaluate the comparison and linear regression based on the water pressure sensor and manometer shown in Figure 8.

From this plot, the linear regression equation is obtained. This equation is then implemented in the code to correct the water pressure values. R square (R^2) or the coefficient of determination value from this test is 0.8846. The strength of the relationship between water pressure sensor data and manometer data is 88.46%. It means that about 88.46% of the variation in one of the variables is explained by the other.

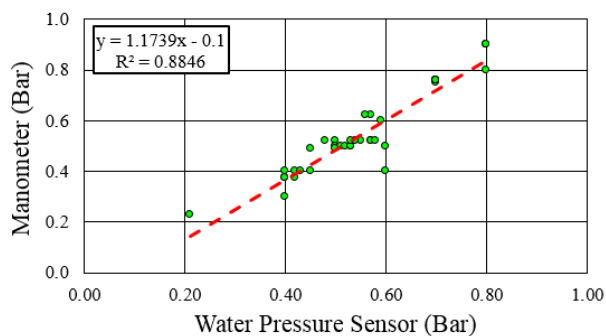


Figure 8. The scatter plot of the water pressure values versus manometer with fitted line and linear regression equation.



Figure 9. All components inside a waterproof case.

C. Prototype Deployment and Experimental Result

This prototype was developed based on the wiring diagram in Figure 2. All components were integrated on a fabricated PCB and placed inside a waterproof case as shown in Figure 9. Water pressure sensor connects to the device using CB connector 5 pins. A DC adaptor is used as main power supply. A fuse is used to protect the wiring. Configuration file (config.txt) saved in a micro SD card is shown in Figure 10.

This prototype was successfully deployed at District Metering Area (DMA) Cipaku PDAM Tirta Pakuan, Bogor, Indonesia. The water pressure sensor was installed in a water pipeline near the primary water tank and connected to an analog manometer as comparison as shown in Figure 11 and Figure 12 respectively.

A PDAM technician can view the data from <http://igauge-logger.mantisid.id>. The technician must insert their username and password as shown in Figure 13 to secure the connection. The prototype then send data to server every 5 minutes. The prototype was reading and sending data from April 10th, 2019 until October 31st, 2019. Water pressure sensor data are displayed in time series using high chart. Users can choose the time range as shown in Figure 14. Users are also able to download pressure data from the server using Data Library menu or using high chart menu.

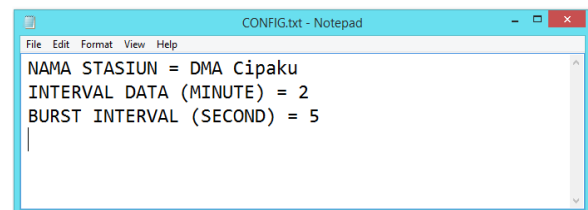


Figure 10. Configuration file.



Figure 11. Water pipe installation to be measured by the prototype.



Figure 12. Sensor installation near analog manometer as comparison.

Based on time stamp from the sensor reading and server time stamp, we are able to obtain the average time in sending the data that is around 30 seconds. The duration of the sending time is acceptable by PDAM users. In addition, only 96% data were sent to server. It is assumed that the occurrence of blank data received by the server is due to issue in operator signal quality.



Figure 13. Login form on water pressure dashboard website.

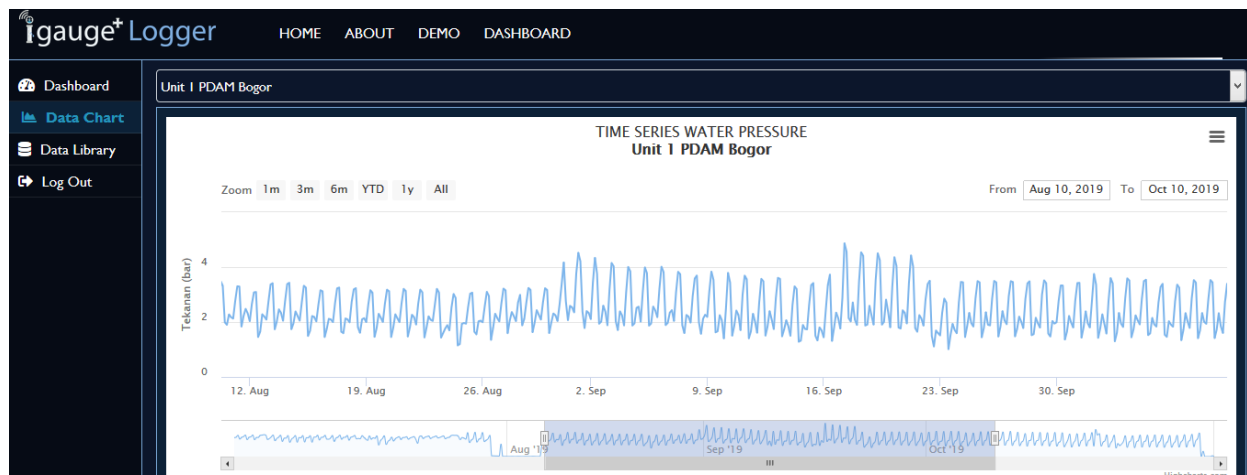


Figure 14. Water pressure data display on real-time dashboard

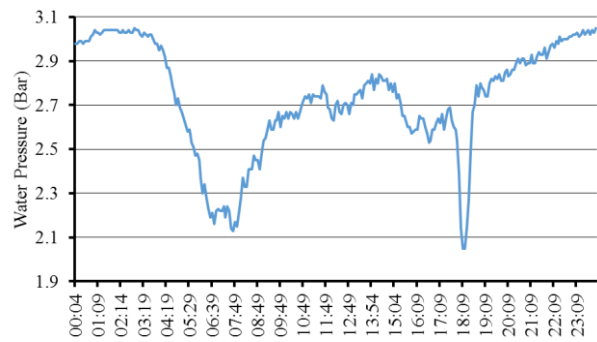


Figure 15. Water pressure data pattern on April 11st, 2019.

The water supply pattern obtained also shows interesting feature. In Figure 15, fluctuation of the pattern is observed. Water pressure drops dramatically at time period 5:29 until 8:04 and at 17:49 until 18:49. This drop occurs due to the water activity of household within this period. In contrast, the water pressure increases until it reaches 3 bar when there is no activity. This phenomenon can be found from 21:00 until midnight. Based on the data, PDAM team must ensure the water supply debit during the busy time period is sufficient until it reaches the users

CONCLUSION

This paper presents the implementation of open source hardware for a real-time water pressure monitoring. The system has been successfully built and tested. Both accuracy and performance tests show acceptable results. With the data interval set to 5 minutes, nearly 100% of the data are obtained. Therefore, the prototype can be used widely in PDAM area to help them monitoring pressure in water pipe line. To improve its usability, future work is required including to develop off grid system to power the prototype so it can be installed in remote area where no PLN electricity available.

ACKNOWLEDGMENT

This research was funded by USAID – IUWASH PLUS Refinement of Prototype (real-time Water Pressure Sensor) for 3 PDAMs (Kota Bekasi, Kota Bogor and Kota Depok) 2019.

REFERENCES

- [1] B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, "Trends and Strategic Researches in Internet of Everything," in *Internet of Everything Algorithms, Methodologies, Technologies and Perspectives*, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springer, 2018, pp. 1–12.
- [2] F. Khodadadi, A. V. Dastjerdi, and R. Buyya, "Internet of Things: an overview," in *Internet of Things*, R. Dastjerdi and B. A. Vahid, Eds. Elsevier, 2016, pp. 3–27.
- [3] D. Serpanos and M. Wolf, *Internet-of-Things (IoT) Systems*, First. Cham: Springer International Publishing, 2018.
- [4] M. Margolis, *Arduino Cookbook*, 2nd ed. Sebastopol: O'Reilly Media, 2012.
- [5] D. Uckelmann, M. Harrison, and F. Michahelles, "An Architectural Approach Towards the Future Internet of Things," in *Architecting the Internet of Things*, D. Uckelmann, M. Harrison, and F. Michahelles, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–24.
- [6] M. Banzi and M. Shiloh, *Make: Getting started with Arduino*, Third. Sebastopol: Maker Media, Inc., 2014.
- [7] A. S. Ali, Z. Zanzinger, D. Debose, and B. Stephens, "Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection," *Build. Environ.*, vol. 100, pp. 114–126, 2016.
- [8] H. Nugra *et al.*, "A Low-Cost IoT Application for the Urban Traffic of Vehicles, Based on Wireless Sensors Using GSM Technology," in *Proc. 2016 IEEE/ACM 20th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2016, pp. 161–169.
- [9] M. M. Rahman, Noor-E-Jannat, M. O. Islam, and M. S. Salakin, "Arduino and GSM based smart energy meter for advanced metering and billing system," in *Proc. 2nd Int. Conf. Electr. Eng. Inf. Commun. Technol. iCEEiCT 2015*, 2015, pp. 21–23.
- [10] K. C. Okafor, G. C. Ononiwu, U. Precious, and A. C. Godis, "Development of Arduino Based IoT Metering System for On - Demand Energy Monitoring," *Int. J. Mechatronics, Electr. Comput. Technol.*, vol. 7, no. 23, pp. 3208–3224, 2017.
- [11] H. Báez, I. Vergara-Laurens, L. Torres-Molina, L. G. Jaimes, and M. A. Labrador, "A real-time flood alert system for parking lots," in *Proc. 2017 IEEE 7th Annu. Comput. Commun. Work. Conf. CCWC*, 2017, pp. 1–5.
- [12] M. N. Rajkumar, S. Abinaya, and V. V. Kumar, "Intelligent irrigation system - An IOT based approach," in *Proc. IEEE Int. Conf. Innov. Green Energy Healthc. Technol. - 2017 (IGEHT 2017)*, 2017, pp. 1–5.
- [13] W. Honghui *et al.*, "Research of the Hardware Architecture of the Geohazards Monitoring and Early Warning System Based on the Iot," *Procedia Comput. Sci.*, vol. 107, pp. 111–116, 2017.
- [14] PDAM Tirta Pakuan Kota Bogor, "Informasi Gangguan," 2020. [Online]. Available: <http://www.pdamkotabogor.go.id/thethme/index.php?m=12&hal=1>. [Accessed: 11-Nov-2020].
- [15] Kementerian PUPR, *Pedoman Pelaksanaan Efisiensi Energi di PDAM*. 2014.
- [16] "DS 3231 RTC General Description," Maxim Integrated, 2015.
- [17] "SIM900 AT Commands Manual V1.11," Shanghai SIMCom Wireless Solutions, 2015.
- [18] R. S. Sedha, *Electronic Measurements and Instrumentation*. S CHAND & Company Limited, 2013.
- [19] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature," *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, 2014.
- [20] A. Gilat and V. Subramaniam, *Numerical Methods for Engineers and Scientists*, 3rd ed. John Wiley & Sons, Inc, 2013.
- [21] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to Probability and Statistics*, 14th ed. Boston: Brooks/Cole, 2013.