ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IOT SYSTEM FOR LEAK DETECTION AND MONITORING OF LIQUEFIED PETROLEUM GAS

[1]FRABOWO PRASETIA, [2]BENFANO SOEWITO

[1,2]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480
E-mail: [1]frabowo.prasetia@binus.ac. id, [2]bsoewito@binus.edu

ABSTRACT

LPG (Liquefied Petroleum Gas) is a common fuel used in restaurants and industries for cooking, heating, etc., but the threat of danger will always be there for every user. Initially, LPG gas does not smell, but if so it will be difficult to detect if there is a leak in the gas cylinder. The very first action is the human sense of smell so that humans can smell the gas. Then, errors also often occur on the manometer used by the regulator is still analog, so the reading of the gas pressure data is still less accurate. The purpose of this paper is to propose an LPG gas leak detection device using MQ2 sensor and LPG gas pressure measurement using MPX5700 sensor, buzzer as an alarm, led, LCD as a gas pressure information media, and serve as a tool to open the regulator valve when a leak occurs. The average results of each measurement with a good distance and time are used is to refer to the limits of excess levels ie more than 100ppm are found in an average condition of 10 seconds with a distance of 10cm, 15cm, 20cm, and 25cm. The Mean Square Error (MSE) value of the MPX5700DP sensor with the gas regulator being compared is 0.09 for the absence of gas conditions received and 0.61 for the gas conditions that have been received by the sensor, this test is to obtain a more accurate value.

Keywords: *Liquefied Petroleum Gas, Gas Leak Detection, Monitoring Gas, MQ2, MPX5700*

1. INTRODUCTION

LPG (Liquefied Petroleum Gas) is a common fuel used in restaurants and industries for cooking, heating, etc., but the threat of danger will always be there for every user [1], [2].

One of the risks of using LPG cylinders is leakage in cylinders or rubber gas so that when exposed to fire can cause a fire. Initially, LPG gas does not smell, but if so it will be difficult to detect if there is a leak in the gas cylinder. The very first action is the human sense of smell so that humans can smell the gas. Therefore, we need a gas leak detection device using Arduino Uno, MQ2 sensor [3], [4].

Then, errors also often occur on the manometer used by the regulator is still analog, so the reading of the gas pressure data is still less accurate. What's more, it is currently sold regulators that use analog manometers but do not have the right unit using only colors and numbers so that to find out the contents of the gas cylinder pressure was not known. Measurement using the MPX5700 sensor is very useful for measuring the pressure of a substance with a range of 15-700 kpa [5]. Therefore, it is necessary to have a digital manometer that can display the actual gas state more accurately.

In this paper, we will propose an LPG gas leak detection device using the MQ2 sensor and LPG gas pressure measurement using MPX5700 sensor, buzzer as an alarm, led, LCD as a gas pressure information media, and servomotor as a tool to open the regulator valve when a leak occurs. The results of gas leak detection and gas pressure measurements will be sent to the cloud as data storage is received by each sensor through the ESP8266-01 module that is connected to Wireless Fidelity (WiFi) so that monitoring can be done at any time via a smartphone.

2. TEORY AND METHODS

2.1 MPX5700

The sensor MPX5700AP sensor is a pressure sensor that has analog or digital input by combining micromachining techniques and bipolar processing to provide accurate signals [6]. MPX5700AP can measure the level of pressure with a range of 15 kPa - 700 kPa or 0-1 101.5 Psi and the analog output results from 0.2 to 4.7 Vdc [7].

2.2 Buzzer

There is a resonance circuit so that a large sound can be obtained. Usually it can be used as a gas alarm, burglar alarm, etc. When gas is detected, the buzzer will make a sound so that it functions as an audio output from the detector [8].

2.3 MQ2

Sensor MQ2 the gas sensor is sensitive to the substance SnO2. This sensor is also very sensitive to LPG, Propane, and Hydrogen and is sometimes used in flammable vapors, very well used for detecting smoke and gas and has a relatively cheap price [9]. The gas sensor is a device that can detect indoor gas, fast response, broad gas detection coverage, sensitivity to LPG, natural gas and City Gas [10].

2.4 Thingspeak

Thingspeak is a web-based open source IoT API platform that can store sensor data of IoT applications with display output in graphical form on the web. Thingspeak can communicate with the internet as a conduit of data connected to the cloud Thingspeak retrieves, stores, analyzes, observes and works on data from sensors that have been connected to Arduino, Raspberry-pi, etc [11].

2.5 MIT App Inventor

App Inventor is an application that can create android applications based on visual block programming, so that in making the application easier for users without the need for any code. Only set and drag-drop blocks [12].

2.6 Esp8266

WiFi esp8266 is a wireless mdul that ca be controlled using two serial ports namely TX and RX. This module can be used to connect WiFi to Arduino to connect to smartphone applications, webservers and digital door locks activated [13].

2.7 System Design

In this system design, there are inputs, processes, and outputs of the system to be made. MQ2 and MPX5700 sensors will be the input of this system, while the *buzzer*, *LCD* 16x2, and *LEDs are* output. When the MQ2 sensor detects gas the *volt* (voltage) of this sensor will increase. data *volt* This is then received by Arduino and then processed to be converted into a gas leak unit (*ppm*). Based on this data, a value that becomes information on gas leakage will be achieved. For the gas leak threshold itself it is targeted to be more than 1500 ppm, the level stated is that the gas content is excessive. *Buzzer* serves as an *alarm* that if the threshold level of gas has exceeded it will sound. The *LEDs* function as outputs if normal gas levels turn green while excess gas levels turn red.

The same thing happened in the design of the MPX5700 sensor as well as the *input* system. The sensor receives pressure from the gas cylinder. The pressure is still in the form of voltage which will be processed by Arduino. Arduino receives the data and is then processed from voltage to a unit of pressure. This pressure data will be used as the information media. Data that has been processed by Arduino will be forwarded to the *LCD* to be displayed as information media.

Besides, the design of this system will be connected to the *cloud* using the module *ESP8266 01* as a connecting media to the network *internet*. After this module is connected to the internet, data from Arduino will be sent to the *cloud* as *database* storage *online*. data *cloud* This is then used on *the smartphone* user's view of pressure and gas leak information.

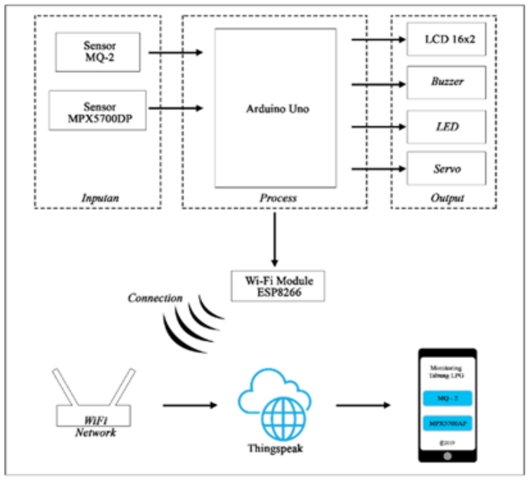


Figure 1: System Design The

Following is a flowchart uploading sensor data to the Thingspeak cloud:

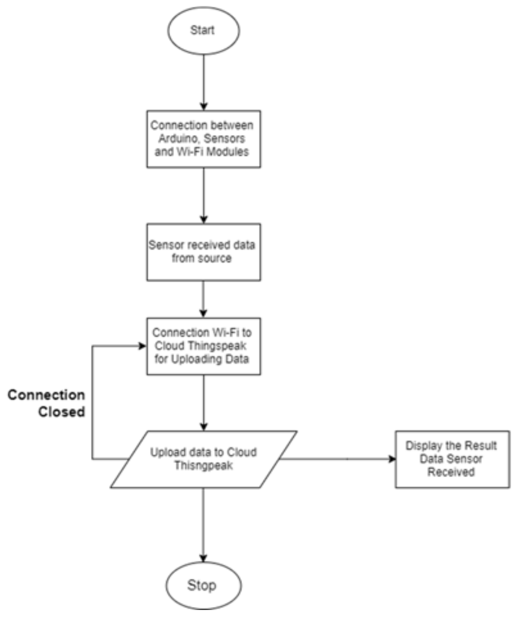

Start
Connection between Arduino, Sensors and Wi-Fi Modules
Sensor received data from source
Connection Wi-Fi to Cloud Thingspeak for Uploading Data
Connection Closed
Upload data to Cloud Thisngpeak
Display the Result Data Sensor Received
Stop

Figure 2: Flowchart Uploading Sensor Data to the Cloud Thingspeak

3. RELATED WORK

In the study Soundarya [14] monitoring system and detection of gas leaks with leak detection of 400-600 ppm, informing the user via SMS, an automatic alarm that is active, and the Exhaust Fan as a protection tool so that gas levels can be reduced from the room. In Sriwati's research [15], using the MQ-6 gas sensor was used to detect LPG, ISO-butane, propane with high sensitivity but, for sensitivity to alcohol and cigarette smoke it was very small to be detected by this sensor. In Naik's research [16], this system in real-time measures the weight of each tube and when the minimum threshold is reached the system automatically sends a message to the official LPG agent and in this study also designs features related to user safety which can detect LPG gas leakage. In Leavline's research, LPG leak detection and warning systems use LEDs and alarm bells to alert users of leaks. This system is designed very simple but provides benefits in handling gas leakage [17]. In Aras's research [18], using two sensors that want to be compared namely MPX4250AP and MPX5700AP. Both of these sensors will be tested at pressures under three different conditions such as in water tanks, lakes, and swimming pools to see the effect. In Pati's research [19], using a gas leak detection sensor. When the sensor detects a gas leak, the microcontroller will activate the stepper motor to open the gas regulator valve, then this will be displayed on the LCD. The use of GSM modules also functions as a media controller automatically.

4. RESULT ANALYSIS

4.1 MQ2 Sensor Testing

In testing this system, the sensor used is the model *MQ2* that can detect LPG gas. This test aims to determine the level of success when detecting gas through the sensor *MQ2* based on parameter *settings time* which has 3 types of time namely 2s, 5s, 10s and adjusts the potentiometer on the sensor *MQ2* with sizes 1/3, 2/3, and 3/3. This test is carried out by employing researchers conducting sensor tests on the tube that leaked through the valve tube. Researchers do this because the sensor is *MQ2* very sensitive to the presence of LPG gas. The origin of the leak by changing the rubber valve tube to provide the effect of leakage so that the process is like the original leak when the gas cylinder. The process of gas leakage is caused by the rubber on the tube not functioning properly so that leakage occurs and the gas that exceeds the oxygen content in the air can easily catch it.

But to get accurate data results by taking data through a hose sourced from the tube. Data collection techniques by connecting the hose to the regulator that has been placed on the valve tube so that gas will flow through the hose that has been connected. For the use of regulators here the researchers used the W181NM model (see figure). Researchers try to rely on hearing aids as an experiment when a sudden leak occurs in a tube with a distance of 5 meters from the leak to do events that often occur. Next, the researchers placed the sensor *MQ2* using a specified distance of 5cm, 10cm, 15cm, 20cm, 25cm, and 30cm. Researchers also hope that these conditions can be read directly by the sensor so that the system directly handles leaks under what has been programmed on the microcontroller.

This information will later be displayed on the *cloud* and withdrawn data for monitoring the situation through a smartphone. The sensor receives a leak in the form of a voltage which is then converted into an ADC (*Analog Digital Converter*) value that functions to convert an analog voltage to a digital voltage.

In the Arduino analog pin it can accept values up to 10bit so that it can represent a voltage of 0 volts and a value of 1023 represents a voltage of 5 volts. Then after the ADC value is obtained, the value is converted into ppm as a unit of gas concentration. *The range* obtained from the sensor *MQ2* has a detection range between 200 - 5000 = 4800, while for the *Total Bits* obtained based on the number of bits from Arduino as much as 1024 including 0. For the detection range can be seen in the *datasheet* attached.

Then from the results *x* above, the measurement process will be carried out *ppm* (part per million)as follows:

$$ppm = 300 + (9.4819159335 * \text{Value } ADC) \quad (1)$$

Obtained several test tables based *on time settings* sensor and potentiometer *MQ2* to optimize system performance. This sensor data retrieval is carried out for 2 minutes with the normalization of the gas after the gas is detected less than 0.20volt 3 times the appearance of the data.

Table 1: Average of 2 Second

Distance	Average 2s (ppm)		
	1/3	2/3	3/3
5cm	681.48	653.77	755.46
10cm	531.23	576.91	608.35
15cm	565.35	525.74	606.52
20cm	542.67	558.09	493.91
25cm	544.30	514.63	404.96
30cm	396.86	365.29	391.24

Table 2: Average of 5 Second

Distance	Average 5s (ppm)		
	1/3	2/3	3/3
5cm	759.75	682.72	764.53
10cm	753.53	759.11	794.35
15cm	809.02	758.63	723.87
20cm	649.55	707.92	549.57
25cm	554.03	455.80	432.20
30cm	432.84	353.74	395.36

Table 3: Average of 10 Seconds

Distance	Average 10s (ppm)		
	1/3	2/3	3/3
5cm	853.85	844.95	979.27
to 10cm	1094.58	957.50	1239.94
15cm	1187.81	1085.08	1085.99
20cm	1115.44	880.53	884.82
25cm	1024.97	589.81	524.79
30cm	656.66	353, 67	452.42

Based on the test results of several things above, such as testing the distance of the gas source to the sensor, testing the *delay* used when a leak occurs are 2s, 5s, and 10s, and the potentiometer settings of 1/3, 2/3, and 3/3 rounds can be concluded that with the use of 1/3 round potentiometers and with sensor distances r to the source of leakage and *delay* that is appropriate to use in the system is a test in which the sensor often captures gas levels of more than 1500 *ppm* because with such a thing the sensor can directly provide *input* to the system that a leak has occurred and what is not recommended is the distance, *delay*, and rotation of the potentiometer that is not often or rarely sensors detect excess gas levels.

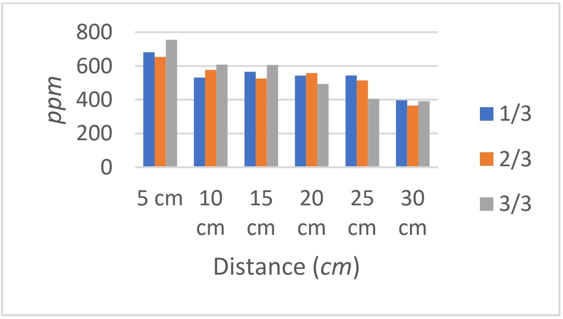


Figure 3: Graph of an Average of 2 Second

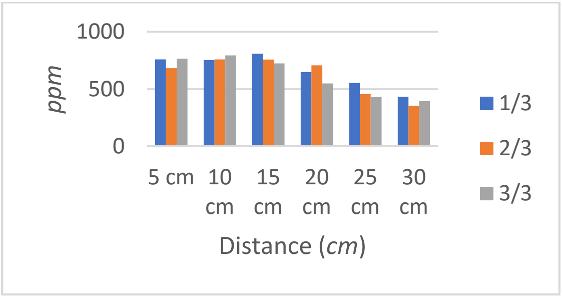


Figure 4: Graph of an Average 5 Second

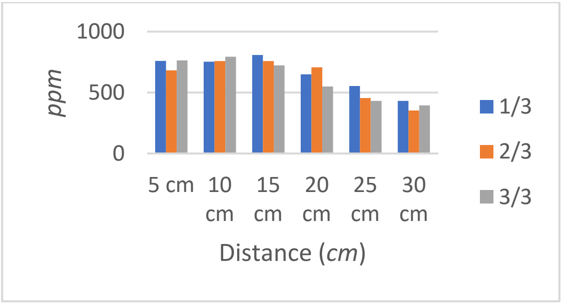


Figure 5: Graph of an Average 10 Second

4.2 Comparison of MQ2 Sensor Results with the GD13 Combustible Gas Detector

Based on the results of the tests conducted above, researchers tried to compare the results of data from sensors with existing tools. Before starting comparative testing, researchers do the data synchronization first by looking at the user guide of the tool. The limit of excessive levels used is 2500 *ppm* with a gas concentration of 300-10000 ppm. The distance used is 5 cm and the time *delay* 2s. At the time of the initial experiment, first tested by a system that is made, the sensor detects the gas then ppm rises to reach the gas content limit then the sensor sounds at 2907.5*ppm*, but researchers want to try to continue detection because the comparative instrument detects gas will continue to rise even though gas is gone. So the researchers tested the system that was made to be able to equalize the results of the comparative tool. For example in a comparator that detects a gas level of 3496 *ppm*, it has been explained before that in a system that is made a continuous gas detection in each of these experiments because to find the value obtained by the gas and the comparator has a similarity or its value is almost close to the results of the comparison. So it takes 3-4 times the ups and downs of gas received by the system created to approach the results detected by the comparator.

Table 4: Comparison of MQ2 Data Sensors with Combustible GD13Gas Detector

Experiment	***Sensor MQ2 (ppm)***	***Combustible Gas Detector GD13 (ppm)***	**Error (Et)**	**Square Error (Et2)**
1	3457.5 ppm	ppm 3464	to 6.5	42.25
2	ppm3476.4	3496ppm	19.5	384.16
3	2803.2 ppm	2544 ppm	-259.2	67184.64
4	3580.7 ppm	3368 ppm	-212.7	45241.29
5	5524.5 ppm	5120 ppm	-404.5	163620.25
Number of Error Squares				276472.59
Value of *Mean Square Error* (MSE)				55294.52

Calculation of the *Mean Square Error value is* firstly performed to calculate an error to get the results of the comparison between the actual value and the results, by reducing the number of actual values with the value of the results. For error formulas based on research [20] can be seen below.

$$Et = Xt - Ft \qquad (2)$$

Note:
Et = error value
Xt = actual data in the t period
Ft = result from data in the t period

After obtaining an error value of the two values then proceed with squaring the error value. The formula for calculating the *Mean Square Error* (MSE) value from the calculation of the acquisition that has been done. *Mean Square Error* (MSE) is a parameter to test the accuracy of the results that have been made. The smaller the value of *Mean Square Error* (MSE), the more accurate the results have been made. Calculation of Mean Square Error (MSE) can be seen below:

$$MSE = \sum Et^2 / n \qquad (3)$$

Explanation:
Et^2 = quadratic error value
n = a lot of data

Based on the table it can be concluded that the *Mean Square Error* (MSE) value from the MQ2 sensor comparison with the GD13 Combustible Gas Detector produces fairly accurate results.

Here is a comparison image of the MQ2 sensor and the *Combustible GD13Gas Detector*.

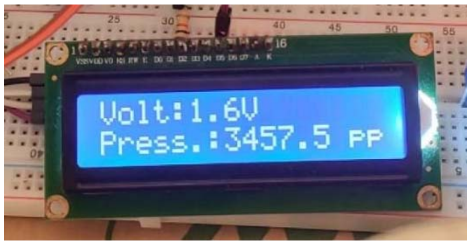


Figure 6: MQ2 Sensor Detection Results

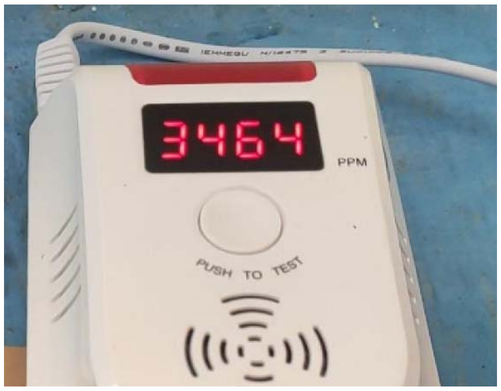


Figure 7: Detection Results from Combustible Gas Detector GD13MPX5700

4.3 Sensor Testing

In MPX5700DP sensor testing is used in measuring the state of the contents of the gas cylinder so that it looks more accurate. Based on researchers' observations of events that occur in the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

community is not known the actual contents of each gas cylinder used at home community.

Currently many regulators are sold that only use manometers in the form of numbers and colors between the numbers one with the next number. This manometer does not also include the meaning of the colors used, whether it is read as a signed color if the contents of the gas in a cylinder *full* of other colors. This event is quite confusing as the LPG gas cylinders whether the contents of the tube are actually filled *full* or only partially. This limitation is what we sometimes ignore, even if when we know the contents of the tube does not match what is listed.

One time researchers tried to weigh the weight of a gas cylinder that had just been purchased. So, the weight of each scale is around 7.8 kg which is 5 kg for the weight of the cylinder and about 2.8 kg for the gas in the cylinder. However, researchers tried to look through the manometer regulator only the colors that were used and did not include in the form of actual units.

Table 5. MPX5700 Sensor Testing Results

Volt	kPa	Psi	bar
0.2	28.089	4.073	0.3
4.5	697.861	101.190	6.9

The test itself is carried out directly by testing the MPX5700DP sensor with 3 kg gas cylinder gas pressure. Before conducting the test, the researchers first made a slight modification to the regulator by replacing the manometer by cutting the manometer to the regulator. After a modification to the Nepel, the researchers connected the hose from the sensor to the nepel so that the gas could be directly measured. For the initial gas state before the pressure, the sensor will get a minimum voltage of 0.2v with a pressure of 0.3 bar. This is stated on the LCD that has been programmed in Arduino.

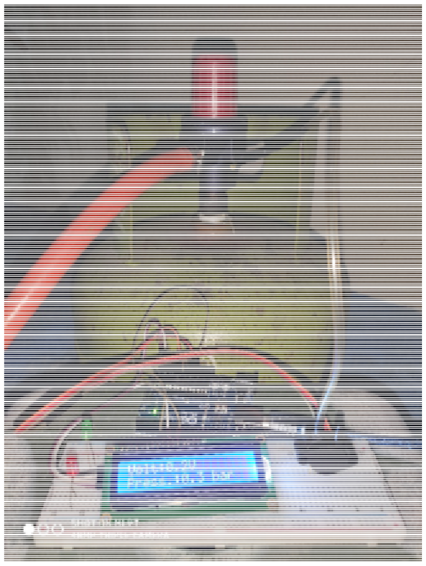


Figure 8: Sensors Receive 0.2v Voltage and 0.3 bar Pressure

When done by turning the regulator valve so that the sensor can receive pressure, the sensor receives a voltage of 4.5v with a pressure of 7 bar or equivalent to 0.7 MPa.

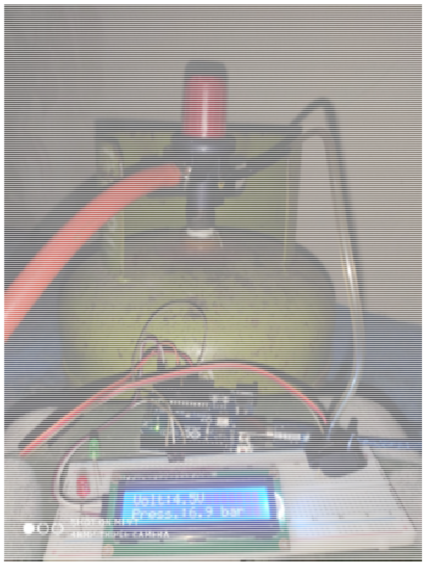


Figure 9. The sensor receives a voltage of 4.5v and a pressure of 6.9 bar.

Researchers here try to compare the results of the system made with existing regulators. There are two regulators as a comparison, the first regulator without specifying units only in the form of numbers with a range of 0 - 10, while the second one regulating with a unit bar (MPa) *range of* 0 - 8 bars. When testing the first regulator, the needle manometer leads to number 7, this indicates that the sensor is made under the results to be achieved.

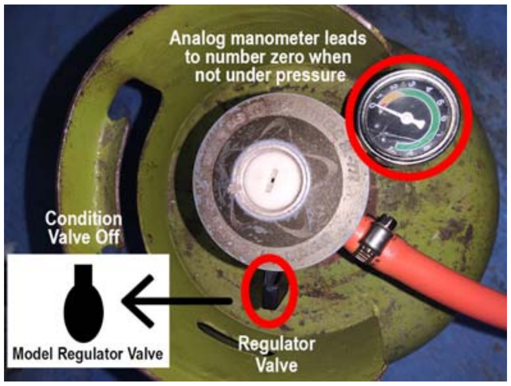

Analog manometer leads to number zero when not under pressure
Condition Valve Off
Model Regulator Valve
Regulator Valve

Figure 10: First Regulator with Valve Not Locking Condition

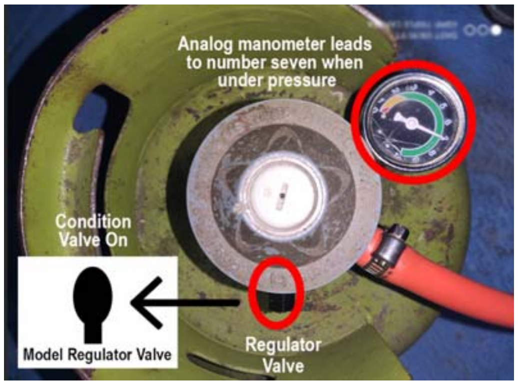


Figure 11: First Regulator with Valve Locking Condition

For the second regulator, the needle manometer leads to number 8, this indicates that there is a difference that is not much different from the results received by the sensor.

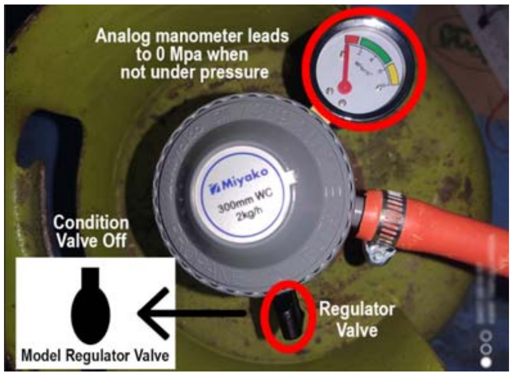


Figure 12: Second Regulator with Valve Condition Not Locking

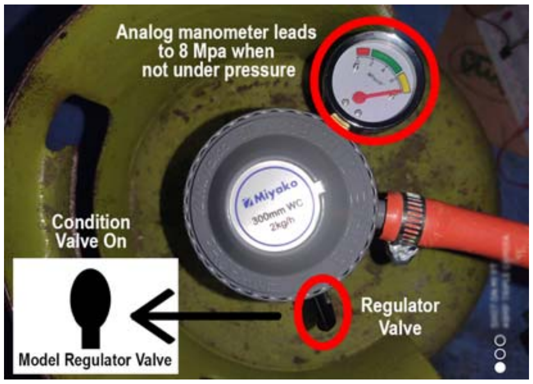


Figure 13: Second Regulator with Valve Condition After Locking

After seeing the condition of the second regulator, the researchers calculated how much error occurred in the sensor made with the second regulator. First the researchers do the above, then take the original data directly from the gas cylinder. Initially this sensor is received is a voltage and then converted into the form of pressure units namely kpa (Kilopascal) and psi (Pounds per Square Inch). Note the equation below that is derived from the thesis of the automatic water rocket launcher:

$$\text{Volt} = V * (5/1023) \quad (4)$$

V here is the result of pin A0 used on Arduino as an analog value reader than on Arduino. This is because the voltages received do not match the voltage 0 - 5 volts. After getting the volt value, it will be converted into a pressure unit. The following is the equation in finding kpa and psi sourced from the thesis of an automatic water rocket launcher contained in the attachment:

$$\text{kPa} = ((\text{Volt} / 5) - 0.04) / 0.0012858 \quad (5)$$

$$\text{Psi} = \text{kpa} * 0.145 \quad (6)$$

On the MPX5700DP sensor datasheet that the pressure range owned by this sensor is 15 - 700 kpa, for 1 kpa = 0.145 psi, while 1 psi according to the barometer produces 6.89 kpa (it can be seen in the attached datasheet). Then the sensor data obtained through the table. As follows:

Table 6. Testing sensor MPX5700DP

No.	First Regulator (Without Unit)	Second Regulator (Unit)	MPX5700 Sensor	Error (Et)	Error Squares (Et^2)

1	0	0 bar	0,3 bar	0,3 bar	-0.3	-0.3	0.09	0.01
2.	7	8 bar	6,9 bar	6,9 bar	0.1	1.1	0.09	1.21
Total Error Squares							0.18	1.22
Mean Square Error (MSE)							0.09	0.61

Can value calculation *Mean Square Error* done first error calculation to obtain the result of a comparison between the actual value and the results, by reducing the number of actual values by the value of the results. For error formulas based on the formula (2).

After getting the error values from both values then proceed with squaring the error values. The formula for calculating the *Mean Square Error* (MSE) value from the calculation of the acquisition that has been done. *Mean Square Error* (MSE) is a parameter to test the accuracy of the results that have been made. The smaller the value of *Mean Square Error* (MSE), the more accurate the results have been made. Calculation of Mean Square Error (MSE) can use formula (3).

Based on *Table 6, it* can be concluded that score Mean Square Error *(MSE) from* MPX5700DP sensor testing with a regulator produces fairly accurate results.

4.4 Thingspeak

Testing This test is carried out so that the data that has been obtained is sent to the cloud that has been connected with the help of the ESP8266-01 module whose function is to connect to the internet/wifi at every home.

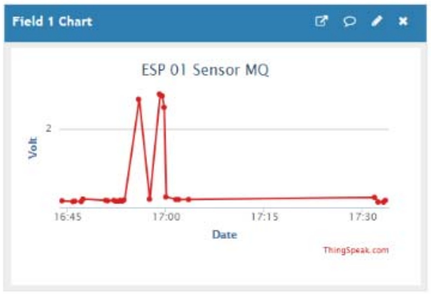


Figure 14: Testing of Thingspeak Cloud It

can be seen that the results of data obtained by the sensor were successfully sent to the *Thingspeak cloud* by making a graph so that researchers more easily analyze the developments that occur in each sensor. In the *cloud there* this is also a *button Export recent data* that works if you want to export it in the form of JSON, XML, and CSV.

4.5 Android App Testing

In this test a smartphone application was tested to monitor the state of the gas both leakage and gas pressure that has been sent to the cloud (Thingspeak). In making this application using MIT App Inventor based on cloud which greatly facilitates researchers in making smartphone applications.

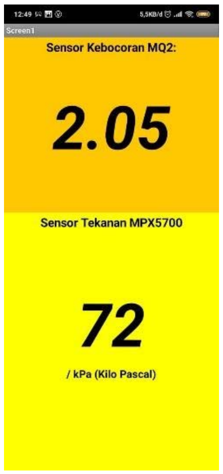


Figure 15: Smart Regulator Application of Gas Leak Detection and Gas Monitoring based on IoT

Seen in "Figure 11" there are 2 pieces of information that can be seen namely the state of the gas leak sensor and gas pressure monitoring. ThingSpeak will provide a continuous signal to applications that have been made and only have a delay of 1 second. Some problems arise when the application sends a response to ThingSpeak, ThingSpeak also continuously sends signals to the application.

4.6 Testing *Buzzer*

The testing *buzzer* is intended to make known whether it can function properly in the event of a gas leak detection with high levels of concentration more than 1000 ppm was programmed in Arduino. In this test, the gas leak detection sensor is not given first so the *buzzer is* off, then the sensor is brought near the source of the gas leak to detect the excess gas concentration. But here are some conditions, when the sensor detects the gas and the gas concentration less than 1000 ppm then the *buzzer* remains off, whereas if the sensor detects the gas and the concentration more than 1000 ppm then the *buzzer* will live, testing is *Buzzer* performed on one of the tables with a distance of 20cm and with a delay of 10 seconds. This indicates that the *buzzer is* functioning properly. Consider the following table.

Table 7: Testing Buzzer

No.	1/3	Buzzer
	ppm	
1	371.76ppm	Off
2	1986.38 ppm	Off
3	379.73 ppm	Off
4	1946.51 ppm	On
5	363.79 ppm	Off
6	1926.58 ppm	On
7	375.75 ppm	Off
8	1934.55 ppm	On
9	371.76 ppm	Off
10	1906.64 ppm	On
11	363.79 ppm	Off
12	1938.54 ppm	On

4.7 Servo

Testing test *servo* This aims to determine whether it can function properly when detection occurs gas leaks with concentrations of more than 1000 ppm that have been programmed in Arduino. In this test, the gas leak detection sensor is not given first so that the gas *servo is* off, then the sensor is brought near the source of the gas leak to detect excess gas concentration. But here are some conditions, when the sensor detects the gas and the gas concentration less than 1000 ppm then the *servo* remains off, whereas if the sensor detects the gas and the concentration more than 1000 ppm then the *servo* will live. This indicates that the *buzzer is* functioning properly.

Table 8: Testing Servo

No	1/3	Servo
	ppm	
1	375.75 ppm	Off
2	375.75 ppm	Off
3	1723.26 ppm	On
4	371.76 ppm	Off
5	1838.87 ppm	On
6	371.76 ppm	Off
7	1878.74 ppm	On
8	387.71 ppm	Off
9	1615.61 ppm	On
10	375.75ppm	Off
11	1802.99 ppm	On
12	363, 79 ppm	Off
13	1842.86 ppm	On

4.8 Testing LED

On testing *LED* this function is to provide a sign in the form of lights. The lights used are as many as two colors namely Green and Red. If the green light is on it means the condition is safe and no leakage occurs but if the condition occurs leakage then the red light will turn on. See the picture below

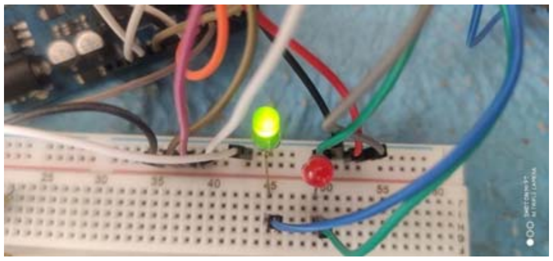


Figure 16: Green Light

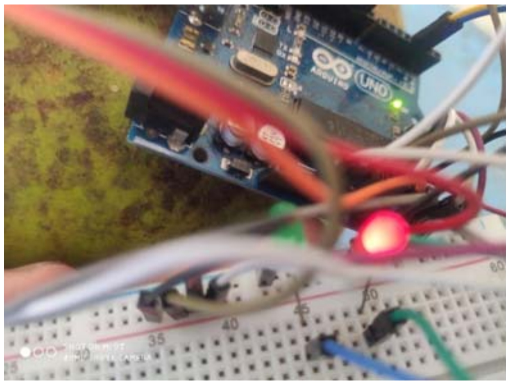
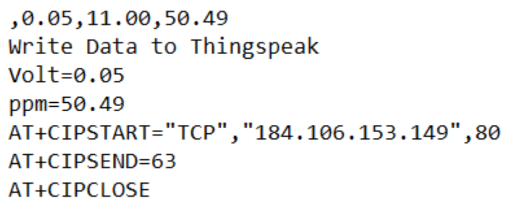


Figure 17: Red Light

4.9 Data to Cloud

In this test, the researchers tested the *delay of* sending sensor data to the *cloud*. This test is carried out to ensure that the data received by the time ordered by the system so that when updating data is very fast and precise.

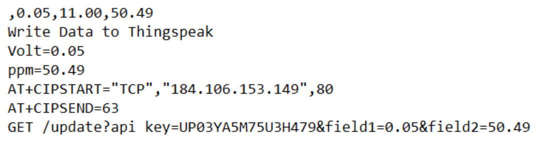
The data is sent directly from the sensor to the microcontroller then reads the system and sends sensor data to the *cloud* using *ESP8266 01* as a link to the available internet network. However, there are problems when sending data to the *cloud* sometimes takes a few seconds to connect to the internet. This is caused due to interference on the internet or an unstable internet connection. Note the picture., The picture displays if the data received from the sensor can not then be sent to the *cloud* marked with "AT + CIPLOSE".



```
,0.05,11.00,50.49
Write Data to Thingspeak
Volt=0.05
ppm=50.49
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=63
AT+CIPCLOSE
```

Figure 18: Data Cannot be Send to the Cloud

If the sending of sensor data to the *cloud is* successful it will be marked with "AT + CIPSEND" and GET / UPDATE? API_KEY = "API Key and its fields".



```
,0.05,11.00,50.49
Write Data to Thingspeak
Volt=0.05
ppm=50.49
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=63
GET /update?api key=UP03YA5M75U3H479&field1=0.05&field2=50.49
```

Figure 19: Data Successfully Sent to the Cloud

5. CONCLUSION

After getting the results and evaluating these results, the conclusions obtained in this study are as follows: This system can detect LPG gas well and is quite accurate. The average results of each measurement with a good distance and time are used is to refer to the limits of excess levels ie more than 100ppm are found in an average condition of 10 seconds with a distance of 10cm, 15cm, 20cm, and 25cm. The user interface on the smartphone is displayed with the last amount of content uploaded using an internet connection that is connected using the esp8266 01 modules. Data is successfully sent to the cloud with a delay of one minute and two seconds. The data was successfully made as information on the smartphone by displaying the level of gas leakage and measurement of LPG gas pressure. The Mean Square Error (MSE) value of the MPX5700DP sensor with the gas regulator being compared is 0.09 for the absence of gas conditions received and 0.61 for the gas conditions that have been received by the sensor, this test is to obtain a more accurate value.

REFERENCES

[1] T. A. S. Raja, R. S. Kumar, A. Nandhakumar and K. V. S. Kumar, "LPG Leakage Detection and Autorefilling," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. VIII, no. 2S, 2018.

[2] A. G. Omar, A. Y. A. Ditual, J. E. Urot, M. R. G. Dimal, N. G. Mamco and C. T. Sagarino , "Mq2-Tector: An Arduino Based Gas Detector, Preventing Gasleak Explosion," 2019.

[3] B. T. W. Utomo and D. S. Saputra, "Simulasi Sistem Pendeteksi Polusi Ruangan Menggunakan Sensor Asap Dengan Pemberitahuan Melalui SMS (Short Message Service) Dan Alarm Berbasis Arduino," *Jurnal Ilmiah Teknologi dan Informasia ASIA (JITIKA),* vol. 10, 2016.

[4] S. Mulyati and Sumardi, "Internet of Things (IoT) pada Prototipe Pendeteksi Kebocoran Gas berbasis MQ-2 dan SIM800L," *Jurnal Teknik: Universitas Muhammadiyah Tangerang,* vol. 7, pp. 64-72, 2018.

[5] F. Ahmad, "Perancangan Alat Ukur Tekanan Udara dengan Menggunakan Sensor Pressure Gauge Mpx5700 Berbasis Arduino Uno," Universitas Sumatera Utara, Medan, 2018.

[6] R. A. Setiawan and D. M. Midyanti, "Rancang Bangun Alat Monitoring Tekanan Angin Ban secara Real Time menggunakan Metode Tsukamoto pada Kendaraan Roda Empat," *Jurnal Coding, Sistem Komputer Untan,* vol. 06, pp. 54-65, 2018.

[7] A. N. D. Mufidah, A. Setyawan, I. Gunadi and J. E. Suseno, "The biodegester flow distribution control system using pressure sensor MPX5700AP," in *Journal of Physics: Conference Series*, 2019.

[8] A. D. Prabhu and A. D. Pathak, "Gas Leak Detector using Arduino Uno Microcontroller," *International Journal for Research in Applied Science & Engineering Technology,* vol. V, no. 7, pp. 1452-1456, 2017.

[9] M. Y. Hariyawan, A. Gunawan and E. H. Putra, "Wireless Sensor Network for Forest Fire Detection," *TELKOMNIKA,* vol. XI, 2013.

[10] M. S. Kolhe and V. Londhe, "Wireless Sensor Network Based Internet of Things For Environmental Impact Analysis," *International Research Journal of Engineering and Technology (IRJET)* , 2018.

[11] S. Pasha, "Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis," *International Journal of New Technology and Research (IJNTR),* vol. II, no. 6, pp. 19-23, 2016.

[12] M. Taufiq, A. V. Amalia and Parmin, "The Development of Science Mobile Learning with Conservation Vision based on Android App Inventor 2," *Unnes Science Education Journal,* 2017.

[13] J. Seo, N. Jeong and J. Kim, "Design and Implementation of Facilty Management System Using IoT Technology of 4th Industrial Revolution," *Journal of Theoretical and Applied Information Technology,* vol. 96, pp. 7824-7833, 2018.

[14] T. Soundarya, J. V. Anchitaalagammai, G. D. Priya and S. S. K. Kumar, "C-Leakage: Cylinder LPG Gas Leakage Detection for Home Safety," *Journal of Electronics and Communication Engineering (IOSR-JECE),* pp. 53-58, 2014.

[15] Sriwati, A. Suyuti, A. Ahmad and A. E. U. Salam, "Intelligent System of LPG Gas Leakage Detection for Web-Based Living House Security," *ICIC Express Letter,* pp. 89-96, 2019.

[16] R. Naik, P. N. Reddy and S. Kishore, "Arduino Based LPG gas Monitoring & Automatic Cylinder booking with Alert System," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* vol. XI, no. 4, pp. 06-12, 2016.

[17] E. J. Leavline, D. A. A. G. Singh, B. Abinaya and H. Deepika, "Lpg Gas Leakage Detection and Alrt System," *International Journal of Electronics Engineering Research,* pp. 1095-1097, 2017.

[18] M. S. M. Aras, S. S. Abdullah and S. S. S, "Investigation and Evaluation of Low cost Depth Sensor System Using Pressure Sensor for Unmanned Underwater Vehicle," *Majlesi Journal of Electrical Engineering,* 2012.

[19] H. Pati, S. Niradi, J. D. T, S. J. S and S. D. G, "Smart Gas Booking and LPG Leakage Detection System," *Journal of Computer Engineering (IOSR-JCE),* pp. 09-13, 2017.

[20] A. A. Gofur and U. D. Widianti, "Sistem Peramalan untuk Pengadaan Material Unit Injection di PT.XYZ," *Jurnal Ilmiah Komputer dan Informatika,* pp. 13-18, 2013.