International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Prototype Design of Landfill Gas Pipe Leak Monitoring System Based on Microcontroller Node MCU ESP8266 with the Internet of Things Method

Abdul Aziz*[1], Amalia Zahra[2]

Submitted: 12/11/2022 **Accepted:** 15/02/2023

Abstract*:* The pipeline system is the main component of the gas distribution system in the waste power plant. In its development, the pipeline installation system requires continuous supervision and maintenance. It helped reduce losses due to incorrect installation conditions. Pipeline damage often occurs, such as the Landfill Gas (LFG) pipe leak and a decrease in the pipeline's steam production level. The authors plan to make a tool to monitor a landfill gas (LFG) pipe leak in a waste power plant. This tool works because there is a gas leak in the pipeline. This tool will inform the officer who will repair the pipe leak. This tool installs at the top of each corner of the pipe, especially in the pipe connection section. The purpose is to examine the connecting part of the pipe since it is vulnerable to leakage. This tool installs on every pipe connection with one sensor. This tool equips a gas sensor that serves to detect gas leaks. The input is gas. If the gas sensor is on, it indicates a leak in the pipe, then the monitoring system is active and knows how much gas vapor exits from the pipe. If no gas enters, the sensor condition is inactive, and the condition will not leak.

Keywords: *Internet of Things, IoT, LFG, Landfill Gas, Landfill Gas Pipe Leak, Microcontroller Node MCU ESP8266, Monitoring, Pipe Leak Monitoring System, Waste Power Plant.*

1. Introduction

The lack of supervision and monitoring of Landfill Gas (LFG) pipe leakage at Waste Power Plants (PLTSa) in Indonesia is a problem that needs attention. It causes many losses and can also reduce the electrical power generated. If officers did not know the gas leak and how many leak points came out, it would be difficult to find the gas leak's location [1]. Several factors cause gas leaks such as poor pipe quality, poor connection quality to gas pipes, and non-functional tools used [2]. If an adequate monitoring system did not install yet at each gas pipe connection, it could not detect the gas leaks directly. It is detrimental to the PLTSa because if a gas pipe leaks and the gas comes out, there will be a decrease in the electrical power generated. Currently, PLTSa in Indonesia has installed surveillance systems such as CCTV or used workers to monitor pipes prone to leaks [3].

However, the CCTV monitoring system is ineffective because the pressure of the methane gas that comes out through the leaking gas pipe is odourless and colourless [4]. Thus, CCTV cannot monitor gas pipes that have leaks. The amount of gas that comes out of it cannot be known, and if humans inhale it, it will be dangerous for health. Based on this background and also because there are not many studies discussing the internet of things (IoT) for the LFG pipe monitoring system, the author has an innovation to create a LFG pipe monitoring system for PLTSa. This LFG pipe monitoring system will detect if there is a pipe leak. If there is a pipe leak, the tool sends data to the officer regarding the last coordinates of the pipe leak. The design of this LFG pipe leakage monitoring system must be able to detect leaks in real-time and be efficient in its use. In designing and manufacturing this monitoring system, supervisors will have a warning feature that is useful for providing real-time warnings when gas pipeline leaks occur [5]. This monitoring system works because if there is a leak in a leaking gas pipe, the sensor will detect a gas leak at a point [6]. Then, send the data from the sensor to the NODE MCU ESP8266 microcontroller for processing. Furthermore, the processed data at the NODE MCU ESP8266 will be sent to the ThingSpeak database via the internet by adjusting the API key address (Autoritation Public Key) of the Things peak application. Then the sensor data is viewed through internet-connected devices in the ThingSpeak application. The results of the Landfill Gas (LFG) pipe leakage monitoring system in this study are expected to be used as a leak monitoring system for Landfill Gas (LFG) pipelines in PLTSa and can function in real-time, effectively, and

[1] *Computer Science Department, BINUS Graduate Program, Master of Computer Science, Bina Nusantara University, Jakarta, 11480, Indonesia ORCID ID : 0000-0002-1145-7140, Email: abdul.aziz005@binus.ac.id*
[2] *Computer Science Department, BINUS Graduate Program, Master of Computer Science, Bina Nusantara University, Jakarta, 11480, Indonesia ORCID ID : 0000-3343-7165-777X, Email: amalia.zahra@binus.edu*
**Corresponding Author Email: abdul.aziz005@binus.ac.id*

efficiently.

2. Literature Review

2.1. NODE MCU ESP8266

NODE NodeMCU ESP8266 is an electronic board based on the ESP8266 chip. The NodeMCU ESP8266 is able to run the functions of the microcontroller and also an internet connection (WiFi). There are several I/O pins. NodeMCU ESP8266 on the NodeMCU ESP8266 can be developed into monitoring and controlling applications for IoT projects [7]. The NODE MCU ESP8266 is also an open-source IoT platform. Programming for NodeMCU ESP8266 is with Arduino IDE, a compiler from Arduino. The development kit uses an extensive programming language to assist in prototyping IoT products [8]. This kit development base is the ESP8266 module, which integrates GPIO (General Purpose Input Output), PWM (Pulse Width Modulation), IIC (Inter-Integrated Circuit), 1-Wire, and ADC (Analog to Digital Converter) in one board. GPIO NODE MCU ESP8266 in Figure 1 has a length of 4.83cm, a width of 2.54cm, and a weight of 7 grams. This board equips with WiFi and Firmware features that are open source [9]. The NodeMCU ESP 8266 has a USB port (mini USB), which makes programming easy. NodeMCU ESP8266 is also a development derivative module of the ESP8266 group of ESP-12 IoT (Internet of Things) platform modules. This module is almost like the Arduino module platform in function and usability. The specifications of the NODE MCU ESP8266 are as follows:

1. This board is based on the ESP8266 serial WiFi SoC (Single on Chip) onboard USB to TTL. The wireless network standard used is IEEE 802.11b/g/n.
2. Tantalum capacitors 100 and 10 micro farad.
3. 3v LDO regulator.
4. Blue led as an indicator.
5. Cp2102 USB to UART bridge.
6. Reset button, USB port, and flash button.
7. 9 GPIOs with 3 PWM pins, 1 x ADC Channel, and an RX TX pin.
8. 3 ground pins.
9. S3 and S2 as GPIO pins.
10. S1 MOSI (Master Output Slave Input) is the data path from the master to the enslaved person, sc cmd/sc.
11. S0 MISO (Master Input Slave Input) is the data path out of the enslaved person and into the master.
12. SK, SCLK from master to the enslaved person, functions as a clock.
13. Pin Vin as input voltage.
14. Built-in 32-bit MCU.

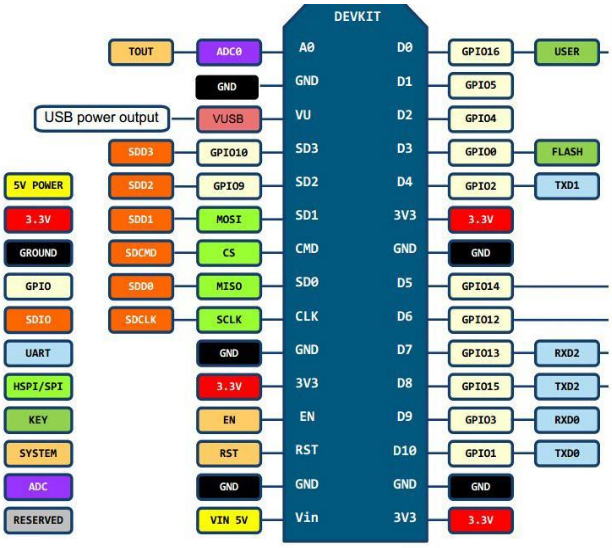


Fig 1. GPIO NODE MCU ESP8266 v3 (createc3d.com)

The function of the GPIO NODEMCU ESP8266 v3 part or component is as follows:

1. RST: function to reset the module
2. ADC: Analog Digital Converter. Input voltage range 0-1v, with digital value range 0-1024
3. EN: Chip Enable, Active High
4. IO16: GPIO16 to wake up the chipset from deep sleep mode
5. IO14: GPIO14; HSPI_CLK
6. IO12: GPIO12: HSPI_MISO
7. IO13: GPIO13; HSPI_MOSI; UART0_CTS
8. VCC: 3.3V (VDD) power supply
9. CS0: Chip selection
10. MISO: Slave output, Master input
11. IO9: GPIO9
12. IO10: GBIO10
13. MOSI: Master output slave input
14. SCLK: Clock
15. GND: Ground
16. IO15: GPIO15; MTDO; HSPICS; UART0_RTS
17. IO2: GPIO2; UART1_TXD
18. IO0: GPIO0
19. IO4: GPIO4
20. IO5: GPIO5

21. RXD: UART0_RXD; GPIO3
22. TXD: UART0_TXD; GPIO

2.2. Gas Sensor

Definition of the gas sensor is a gas detection device that can detect gas based on its aroma. The gas sensor detects pollutant gas compounds in the air, such as carbon monoxide, hydrocarbons, and nitrous oxide. Gas detectors vary and have different levels of sensitivity. The more sensors, the system's sensitivity is very good to various kinds of aromas. If the number of sensors used increases, the price will also be higher. Hence, the more expensive the gas sensor, the better and higher the sensitivity level.

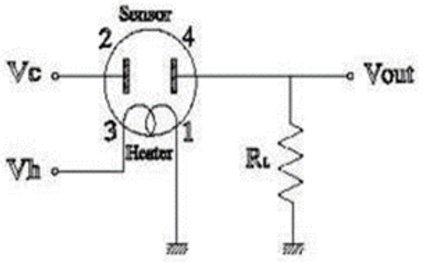


Fig 2. Gas sensor structure (eprints.polsri.ac.id)

Fig 3. Gas sensor (bananarobotics.com)

Figure 2 is the circuit structure of the gas sensor. There is a heater in the circuit structure of the gas sensor. The heater on the gas sensor triggers the sensor to detect the expected gas target after being given a 5V voltage. So two metal elements (2 and 4) will work. There is a predetermined space between the two metal elements. If a sensor detects gas, the density of the space between metals 2 and 4 will increase/decrease. When the resistance gets smaller, the current will flow from 2 to 4, so the sensor voltage output will be large.

The Landfill Gas (LFG) pipe leak monitoring prototype uses the MQ-4 Gas sensor type. The MQ-4 gas sensor type is an electronic component that can detect levels of compressed natural gas or CNG (Compressed Natural Gas), which mainly contains methane gas (methane, CH4), the simplest form of hydrocarbons. Although not toxic, methane gas can be dangerous because it is flammable (combustive/flammable gas). This gas is odorless and colorless, making it difficult to detect directly by humans [10].

2.3. LED (Light Emitting Diode)

Light Emitting Diode (LED) is an electronic component emitting monochromatic light. Monochromatic light on the LED will emit if a forward voltage is applied. LED is also an indicator light for electronic devices that indicates electronic device status [11]. For instance, LEDs are the Power LED on the CPU, the LED on the power supply, the processor indicator LED on the computer, the power saving LED on the monitor, and the LED on the power outlet cable. LEDs are a family of diodes made of semiconductor materials which can light up when an electric voltage of about 1.5 volts DC is applied [11].

LEDs have different characteristics depending on the color they produce. The colors of the light emitted by LEDs depend on the semiconductor used. The higher the current flowing in the LED, the brighter the light produced. However, please note that the permissible current is 10mA - 20mA at a voltage of 1.6V - 3.5V based on the color produced [11]. If the current flowing in the LED is more than 20mA, the LED will burn [12]. Each color in an LED has a different wavelength and semiconductor compound. The following is a table regarding color variations in LEDs:

Table 1. LEDs

Semiconductor Material	Wavelength	Color
Gallium Arsenide (GaAs)	850-940 nm	Infrared
Gallium Arsenide Phosphide (GaAsP)	630-660 nm	Red
Gallium Arsenide Phosphide (GaAsP)	605-620 nm	Orange
Gallium Arsenide Phosphide Nitride (GaAsP:N)	585-595 nm	Yellow
Aluminium Gallium Phosphide (AlGaP)	550-570 nm	Green
Silicon Carbide (SiC)	430-505 nm	Blue
Gallium Indium Nitride (GaInN)	450 nm	White

LEDs are also used in communication on fiber optics as a light source that serves as an information carrier. The average speed of LEDs in carrying information is 50 Mb/s on multimode optical fiber [11]. The advantage of the LED does not require a series of stabilizers for optics when operating the LED. The light emitted by LEDs has little temporal and spatial coherence, which means that LED light has a narrow frequency spectrum [11]. The advantages of using LEDs are low power, more durable, and faster in the switching process [13].

2.4. Buzzer

The buzzer is an electronic component that converts electrical vibrations into sound vibrations [14]. The working principle of a buzzer is almost the same as a loudspeaker [14]. The buzzer consists of a coil attached to a diaphragm, and current flows through the coil to become electromagnetic. Furthermore, the coil will be attracted in or out depending on the current's direction and the magnet's polarity. It happens because the coil mounts on the diaphragm, so every movement will move the diaphragm back and forth so that the air vibrates and produces sound [14]. Equipment often uses a buzzer, such as anti-theft devices, alarms on watches, doorbells, reverse warnings on trucks, and devices for hazard warnings. The advantages of the Piezoelectric Buzzer are that it is cheaper, relatively lighter, and easier to incorporate into other electronic circuits [15]. The piezoelectric effect was discovered first by two French physicists, Pierre Curie and Jacques Curie, in 1880. Then, the development of piezoelectric effect by a Japanese company, and the name changed to the Piezo Electric Buzzer. Piezo Electric Buzzer has been popular since the 1970s [15]. Piezoelectric Buzzer is a type of buzzer that uses the Piezoelectric effect to produce sound or sound. The electric voltage applied to the piezoelectric material will cause mechanical movement [16]. Then, the mechanical movement converts into sound by using a diaphragm and resonator so the human ear can hear it [16].

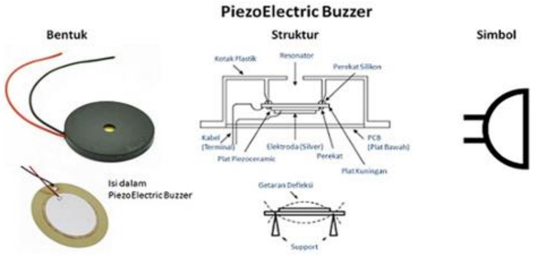


Fig 4. The basic shape and structure of the piezoelectric buzzer (teknikelektronika.com)

Piezo Buzzer can function in producing frequencies in the range of 1 kHz - 100 kHz. Piezoelectric Buzzer Operating Voltage is generally 3 Volts -12 Volts. There are two legs on the buzzer, positive and negative, commonly called transducers. The workings of this transducer are to provide positive and negative voltages of 3 V – 12 V. Then the buzzer will function when electricity or voltage flows into the circuit that uses the piezoelectric [16].

2.5. Methane Gas

Methane is the simplest type of hydrocarbon and is a gas. Because methane is a gas, it is called methane gas. Methane gas is a hydrocarbon type with one carbon atom and four hydrogen atoms and has the chemical formula CH4 [17]. Pure methane gas has no aroma, and if for commercial purposes, it can add a little sulfur smell to detect the leak of methane gas [17]. As the main component of natural gas, methane is the main fuel source. Combustion of one molecule of methane with oxygen will release one molecule of CO2 (carbon dioxide) and two molecules of H2O (water) [17]:

$$CH_4 + 2\,O_2 \longrightarrow CO_2 + 2\,H_2O \tag{1}$$

Methane is one of the greenhouse gases. The concentration of methane in the atmosphere in 1998, expressed as a mole fraction, was 1,745 nmol/mol (parts per billion), up from 700 nmol/mol in 1750. By 2008, methane content in the atmosphere had increased again to 1,800 nmol/mol [18].

2.6. Visual Studio Code

Visual studio code known as VS Code, is a text editor. VS Code is used to write coding or source code [19]. Microsoft created VS Code for multiplatform operating systems. Multiplatform is available in various versions and can operate on Linux, Mac, and Windows platforms [19]. VS Code is a text editor that can write programming languages such as Java Script, Typescript, Node.js, C++, C#, Python, Go, and Java [19]. The features provided by Visual Studio Code, such as Intelligence, Git Integration, Debugging, compiler, SDK, Integrated Development Environment (IDE), documentation (generally in the form of MSDN Library), and extension features to add to the capabilities of the VS Code text editor. These features will continue to grow along with the development of Visual Studio Code versions.

2.7. Black Box Testing

Black box testing is a method of testing the quality of software. The focus on black box testing is testing the software's functionality instead of the internal structure or work [20]. It aims to obtain a series of input and output conditions in accordance with the functional requirements of a program, system, or application [21]. Black box testing also aims to find functional errors, user interface errors, errors in data structures, performance optimization errors, and initialization errors [22]. Black Box testing is also a test of fundamental aspects of the system without regard to the internal logical structure of the software [22]. This method determines whether the software can function adequately [23]. Black Box testing is a test data design method based on software specifications. The testing can run or execute units or modules in black box testing. Then observe the results of the unit or module to determine whether it is the same as the desired process. The purpose of the black box testing is to find the following errors [23]:

1. Wrong, incorrect, and missing functions.
2. Interface Error.
3. Error in a data structure or external database access.
4. Performance Error.
5. Initialization and termination errors.

The use of methods in Black Box testing consists of 3 parts as follows [23]:

1. Graph-based testing.
2. Equivalence Partitioning.
3. Boundary Value Analysis.

2.8. Prototype Method

The definition of the prototype method is a research method used to assist in the process of designing and developing software [24]. The prototype method also has uses such as presenting an initial picture and idea, designing an initial design, looking for problem identifications, and finding solutions to solving these problems. The prototype model has systematic and iterative stages [25] It serves to find out what the stages of making the system are so that the system can be able to operate correctly. The application of the prototype method is to get a representation of the application model or system to be created [25]. The user will re-evaluate the mockup. After users evaluate the mockup, the next stage is that the mockup will be used as reference material for software developers to build applications and produce them ideally. The prototype method is not a perfect development method, but it must be evaluated and modified again since the prototype cycle is iterative. This cycle aims to be able to continue to develop the application or system in the future. Figure 5 below is a schematic or flowchart of the prototype method.

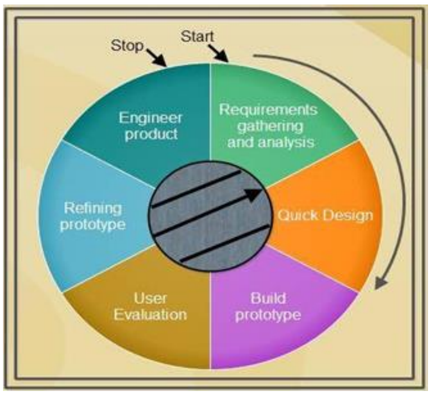


Fig 5. Prototype method steps (ecomputernotes.com)

The prototype method has several stages of research or development as follows [25]:

1. Requirements gathering and analysis

Users will determine and identify the system development requirements. System users, such as admins and users involved in the system, will also be determined.

2. Quick design

This initial design is in the form of a design mockup that refers to the needs of the user or users.

3. Build prototype

The user explains his desire to the system designer to make a temporary design, including quick and easy menu features and input and output displays.

4. User evaluation

The user checks the final build of the Prototype. Checks to ensure that the system is the same as the goals and needs of the user. If the prototype built is different from the user's requirements. There were corrections and improvements to the Prototype's final design. This stage will improve the appearance of input and output that are not appropriate or need to add new features.

5. Refining prototype

The approved Prototype at the previous stage by the user will begin to be processed for manufacture. In making it is by typing the code at this stage, coding using a variety of programming languages. The programming language is generally web-based and is PHP, and the database is MySQL.

6. Product engineer

The system converted into a programming language and has become software will be tested first. Tests are to determine the feasibility of the software. The tests are to minimize errors that exist. The test can use Black Box testing or White box testing. At the product engineer stage, there is also a system evaluation stage. At the evaluation stage, the user evaluates to ensure whether the program or system built is the same as the requirements or not. If it is appropriate, the system can be used. However, if it is inappropriate, the developer must return to the previous stage to correct the discrepancy according to the user's requirements.

After all stages are implemented, the system or application is ready for use. It means that the system built has successfully passed the evaluation stage and can run well.

3. Research Methodology

3.1. The Framework or Steps of Research

There are stages of research to be conducted. The explanation of the stages of the research is in the form of a framework of thought. The framework for developing a prototype Landfill Gas (LFG) pipe leakage monitoring system aims to complete the manufacture of this system systematically and produce maximum results. Figure 6 shows the framework of thought contained in this research.

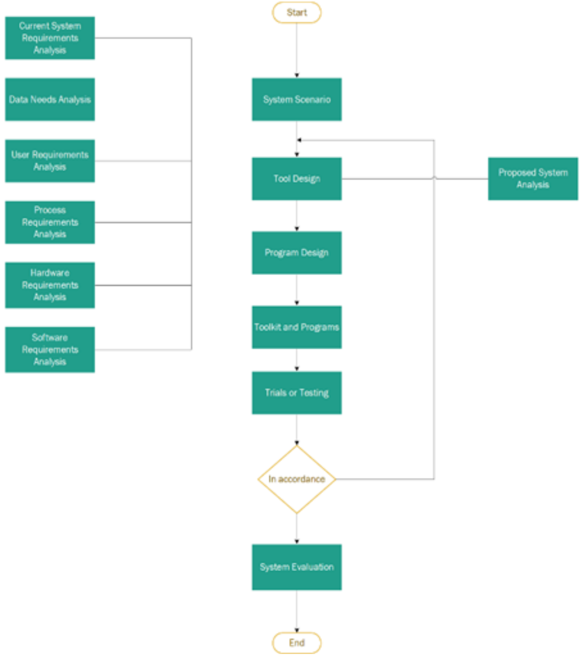


Fig 6. Framework or the steps of research

The explanation of the Framework in Figure 7 is as follows:

1. System scenario

There is a process of identifying the problems contained in the scope of the research. The problem to be solved is to create a Landfill Gas (LFG) pipe leakage monitoring system. After developing the monitoring system, the next step is the implementation of the monitoring system. Development of a monitoring system using the prototype method and the NODE MCU ESP8266 to assist the work of the Landfill Gas (LFG) pipe leakage monitoring process.

2. Tool design

It involves a process of preparing hardware and software components. The prepared hardware is the tool used to develop the monitoring system. At the same time, the software is to support the running of the hardware. The hardware consists of the MQ-4 gas sensor, LED, and buzzer (alarm), which connect to the NODE MCU ESP8266 microcontroller.

3. Software design

There is a process to connect the sensor with the ESP8266 MCU NODE. Connecting these sensors is to detect different Landfill Gas (LFG) pipe leaks. Monitoring is performed from several different pipe leak connection points. The application contains information about gas pipe leaks and will appear on the LED and buzzer indications.

4. Toolkit and programs

The process of assembling tools and programs is combining all tool components and experimenting with each coding of the tool components. The components used in the prototype run using the code entered in the visual studio code application.

5. Testing

After designing the system and application, then a trial is conducted. The trial aims to determine the suitability of the system with system requirements. Suppose it is inappropriate, a system redesign to correct system errors. That must check the system as the same as a requirement. The testing method used is black box testing. Blackbox testing is a functional test based on the system. After testing and finding the error, it will return to the system analysis stage.

6. System evaluation

It creates a test report and documentation of the system, tools, and applications.

3.2. Requirements Analysis

3.2.1. Data requirements analysis

1. The Data of monitoring points of gas pipe leakage.

The number of monitoring points for gas pipeline leaks is an input parameter for the monitoring system, and this displays some valuable data related to gas pipeline leaks. It was conditioning the value of gas pipe leakage, which is still said to be normal or safe and conditioning the value of gas pipe leakage, which is already in a dangerous or unsafe situation.

2. Monitoring time.

The process of monitoring this gas pipeline leak must be in real-time. The monitoring process is data from the MQ-4 Sensor, which is processed and sent to the NODE MCU ESP8266 Microcontroller for processing. Then the processed data from the NODE MCU ESP8266 will be sent to the ThingSpeak database via the internet by adjusting the address of the API key (Autoritation Public Key) in the ThingSpeak application. Furthermore, sensor data can be viewed through internet-connected devices with the ThingSpeak application.

3.2.2. Software requirements

The software requirement for monitoring gas pipe leakage is Visual Studio Code for coding the tools and Thingspeak database to store the data.

3.2.3. Hardware requirements

The hardware requirement for monitoring gas pipe leakage such as:

1. One laptop unit with Processor Intel(R) Core (TM) i3-7250MU CPU up to 3.4GHz RAM with a capacity of 4 GB.
2. 3 Units NODE MCU ESP8266.
3. 3 Units of Gas Sensor Type MQ-4 (Methane Gas).

4. 3 Active Buzzer Units.
5. 3 green LEDs
6. 3 Red LEDs.
7. 2 Mini breadboards
8. 3 Micro USB cables.
9. 2 pieces of pipe (15 cm per piece).
10. 2 pieces of Pipe connection (Elbo).
11. Jumper Cable.
12. Acrylic.

3.2.4. User analysis

Users targeted in systems research monitoring of leakage of Landfill Gas (LFG) pipes at the community mini electric power plant (PLTSa), which in this case is for use by operators and employees at the Community Electricity Local Waste Processing Site (TOSS-LK) at IT-PLN Jakarta.

3.3. Research Design

The author's research will use the prototyping method because the approach describes the flow of the tool to be made, the linkages between components and prepares system requirements in the form of design. The Landfill Gas (LFG) pipe leakage monitoring system integrates with computers and mobile phones.

3.3.1. Prototyping stages

The method of developing a Landfill Gas (LFG) pipe leakage monitoring system at a waste power plant (PLTSa) is the prototyping method. The stages of the prototype method are as follows:

1. System scenario

A system scenario is needed to design a gas leak detector model in the PLTSa pipeline. The system scenario is the initial concept of system development. The tool model must be more efficient and effective than the previous system. It requires a User-friendly interface design to develop the gas leak monitoring applications.

2. Tool design

Tool design is the stage of designing a tool based on a predetermined initial concept. In this tool's design, the hardware components into a leak detection tool in the gas pipeline in PLTSa. The tool can detect the gas leak's location and send notifications to the user. The ESP8266 MCU NODE is the brain of this tool which connects to every component.

3. Program design

After assembling the tools, the program design phase begins. At this stage, it focuses on creating program scripts for tool performance. The script in the build includes a command on the gas sensor to detect the point of a gas leak. NODE MCU ESP8266 will automatically read the leakage scale's value at the pipe connection point, then send notifications to users. The monitoring range is very wide with an internet connection. The tool's working system depends on the ESP8266 MCU NODE coding and the ThingSpeak interface through a computer or a mobile phone.

4. Tool design with programs

This stage aims to assemble and connect the tools and scripts, which is the longest because it requires many experiments to connect the script with the tool properly and optimally.

5. Trial or testing

The next step when the system is complete is testing the system. In addition to testing in the form of software, testing the architecture/structure is also required. In this study, testing is using a black box testing model.

6. System evaluation

After the trial or testing step, it knows deficiencies in the existing system and existing designs to be repaired or completed as desired at the initial thought stage. After it was deemed sufficient, then it was time to limit the problem from the component design of this Landfill Gas (LFG) pipe leak detector model.

3.3.2. Proposed system analysis

In the proposed system analysis, there is a proposal for a system to be designed. The proposed system design and tools to be designed are as follows:

1. Build a series of tools that can assist in checking the Landfill Gas (LFG) pipeline in PLTSa.
2. Create systems and tools that can facilitate the work process of checking Landfill Gas (LFG) pipes when done by one employee.
3. Create systems and tools that provide an alternative for checking Landfill Gas (LFG) pipes in a short time and real-time.

The system analysis design in this study is in the flowchart of Figure 7 System Analysis Design. Figure 7 below is a flowchart of the system analysis design.

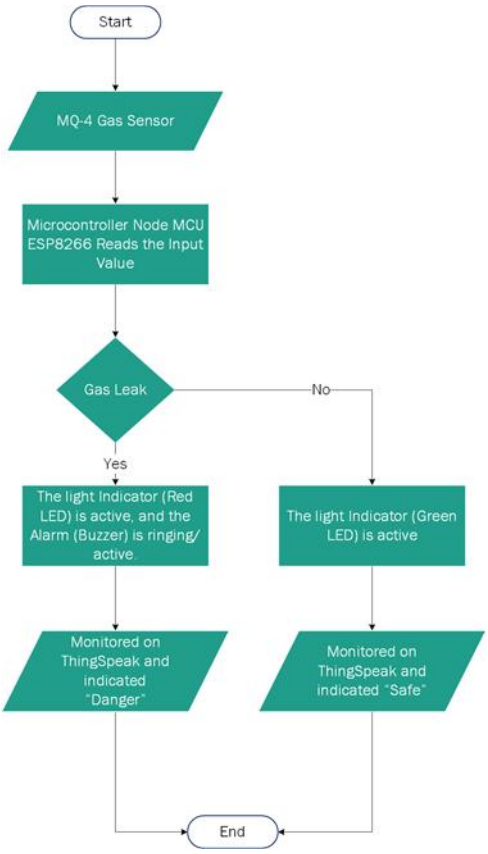

Start
MQ-4 Gas Sensor
Microcontroller Node MCU ESP8266 Reads the Input Value
Gas Leak
No
Yes
The light Indicator (Red LED) is active, and the Alarm (Buzzer) is ringing/ active.
The light Indicator (Green LED) is active
Monitored on ThingSpeak and indicated "Danger"
Monitored on ThingSpeak and indicated "Safe"
End

Fig 7. System analysis design

An explanation of Figure 7 System Analysis Design is as follows:

1. Preparation of the MQ-4 gas sensor to detect leaking gas
2. Microcontroller Node MCU ESP8266 reads the input value
3. Detect gas leaks
4. If a gas leak is detected, it is monitored on ThingSpeak and indicated as "Danger". Next, the indicator light (Red LED) is On, and the Alarm (Buzzer) is Beeping/On. Next
5. If no gas leak is detected, monitor ThingSpeak and indicate "Safe". Next, the Indicator Light (Green LED) is On.
6. Then, the leak detection process is complete. However, the sensor will always work and will automatically detect gas leaks that will occur. The process for detecting it will continue to run according to the flow in the system analysis design unless there are problems.

3.3.3. Hardware design

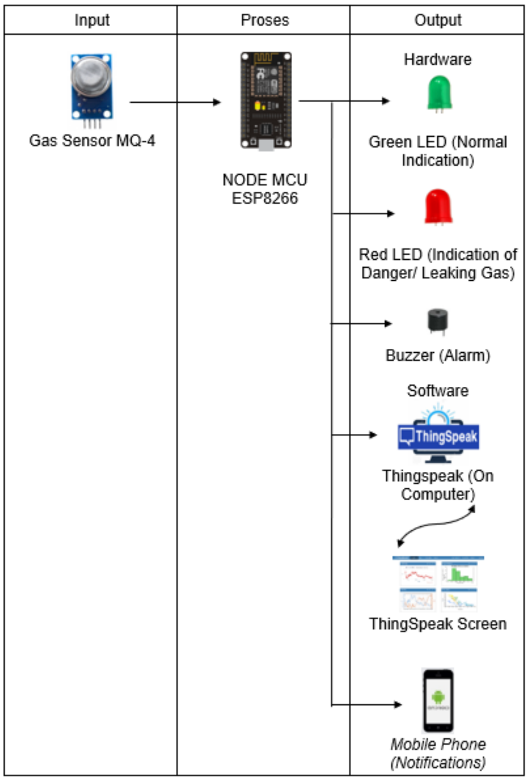


Fig 8. Hardware design

The hardware design in this research consists of combining several hardware components. The hardware components in this study can be in table 3. Table 3 contains the main components for compiling a prototype series of Landfill Gas (LFG) pipe leakage monitoring systems at a waste power plant. This monitoring system combines with the NODE MCU ESP8266 microcontroller and the internet of things (IoT). The hardware design table contains input, process, and output components. The explanation of table 3 regarding the input, process, and output components in the pipe leakage monitoring system is as follows:

1. MQ-4 Gas Sensor

The MQ-4 gas sensor detects the methane gas element from the gas pipe..

2. NODE MCU ESP8266

The ESP8266 MCU NODE is a central control system (processor) of the Landfill Gas (LFG) pipe leakage monitoring tool. The MCU node takes data from the MQ-4 Gas sensor and performs processing of that data to control the LED and buzzer. LEDs and Buzzers are indicators at the location of the Landfill Gas (LFG) pipe installation (control room). The indicator will be sent via the internet network to the ThingSpeak application, then translated into a warning or information humans can understand.

3. LED

The LED (Light Emitting Diode) functions as an indicator in the control room at the gas pipeline installation location as an alternative for monitoring Landfill Gas (LFG) pipeline leaks in addition to monitoring from remote devices.

4. Buzzer

The buzzer is a warning alarm if there is an indication of a leak in the Landfill Gas (LFG) pipe installation.

5. ThingSpeak

ThingSpeak is an application that displays data or information provided by the NODE MCU ESP8266 in real time and can report data on the condition of the Landfill Gas (LFG) pipeline installation.

6. Mobile phones

The Mobile Phone functions as an additional media or alternative for monitoring the condition of the Landfill Gas (LFG) pipe leakage with the ThingSpeak application base and is also able to provide real-time notifications of the condition of the Landfill Gas (LFG) pipe.

3.3.4. Hardware Combination of microcontroller, ESP8266 NODE MCU, LED, Buzzer, and MQ-4 Gas Sensor

The design of the NODE MCU ESP8266 microcontroller uses quite a lot of components. That is because this prototype design requires 3 Microcontroller NODE MCU ESP8266. A microcontroller is a monitoring tool for Landfill Gas (LFG) pipe leakage monitoring. The three microcontrollers will be placed or located at 3 points of the pipe installation connection. Each NODE MCU ESP8266 assembled has the same wiring installation on each pin and part. NODE MCU ESP8266 microcontroller circuit also combines with other components. Figure 8 is the design of the ESP8266 NODE MCU Microcontroller Circuit, LED, Buzzer, and MQ-4.

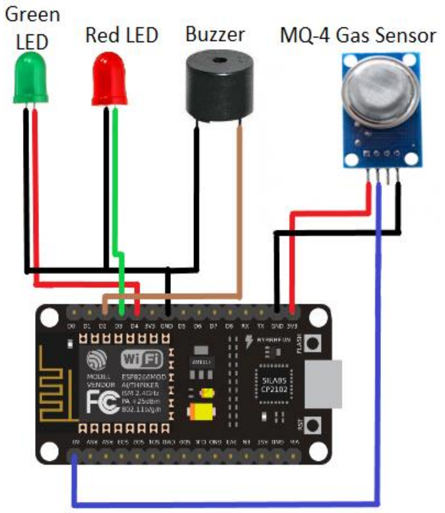


Fig 9. ESP8266 NODE MCU microcontroller circuit, LED, buzzer, and MQ-4

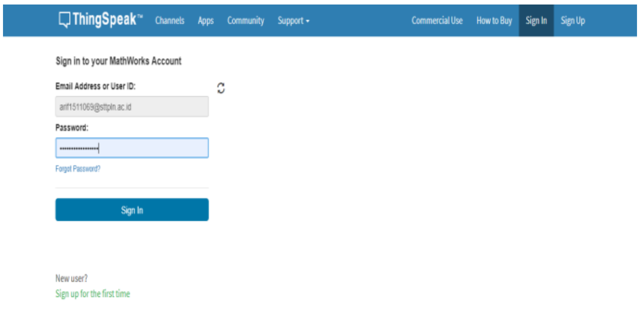
In the final design of the monitoring system tool, combine all the available components. These components connect with cables that have different colors. Each NODE MCU ESP8266 assembled has cable installation on each pin and part. Table 3 describes the details of a series of gas leak monitoring tools.

Table 2. Node MCU ESP8266 microcontroller configuration, LED, buzzer and MQ-4

NODE MCU ESP8266	Gas Sensor MQ-4	Green LED	Red LED	Buzzer	Cable Color
D4 PIN		Anoda (+)			🟥
D3 PIN			Anoda (+)		🟩
D2 PIN				Positif (+)	🟩
A0 PIN	*Analog Output* PIN				🟦
3V3 PIN	PIN VCC				🟥
GND PIN	GND PIN	Katoda(-)	Katoda (-)	Negatif (-)	⬛

3.4. Software Design

3.4.1. Hardware Combination of microcontroller, ESP8266 NODE MCU, LED, Buzzer, and MQ-4 Gas Sensor



Sign in to your MathWorks Account

Email Address or User ID: arif1511069@sttpln.ac.id

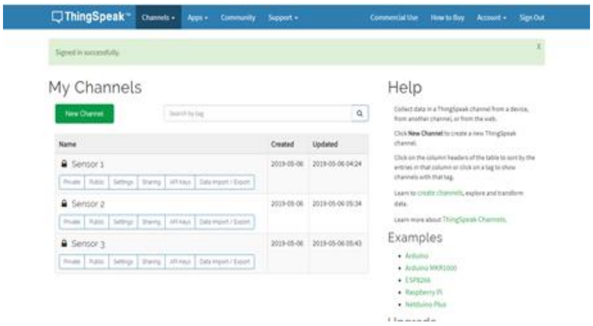
Password: ••••••••••••••••

Fig 10. ThingSpeak login page

Before entering the ThingSpeak database application, log in to the ThingSpeak application first. Figure 9 is the ThingSpeak login page. It Can fill userID and password on the login page to log in to the ThingSpeak application. After logging in, the monitoring page will appear. The

ThingSpeak application requires an internet connection so that the system and application can run and synchronize with the monitoring tools.

3.4.2. ThingSpeak channels view



ThingSpeak™ Channels ▾ Apps ▾ Community Support ▾ Commercial Use How to Buy Account ▾ Sign Out

Signed in successfully. ×

My Channels

New Channel | Search by tag 🔍

Name	Created	Updated
🔒 Sensor 1 Private | Public | Settings | Sharing | API Keys | Data Import / Export	2019-05-06	2019-05-06 04:24
🔒 Sensor 2 Private | Public | Settings | Sharing | API Keys | Data Import / Export	2019-05-06	2019-05-06 05:34
🔒 Sensor 3 Private | Public | Settings | Sharing | API Keys | Data Import / Export	2019-05-06	2019-05-06 05:43

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

Fig 11. My channels page

After logging in, it appears on the Channel page. Figure 10 is a channel page view. The prototype has three sensor points installed on three sections of the landfill Gas (LFG) pipe connection. Therefore, select which part of the sensor to monitor. Choose the part to be monitored by clicking on the sensor text, for example, sensor 1, sensor two, or sensor 3.

3.4.3. User interface design

In software design, the appearance of the software is made simple, attractive, easy to understand, and easy to operate. Every function or use of features in the software is also adjusted to make it look simple and easy to understand.

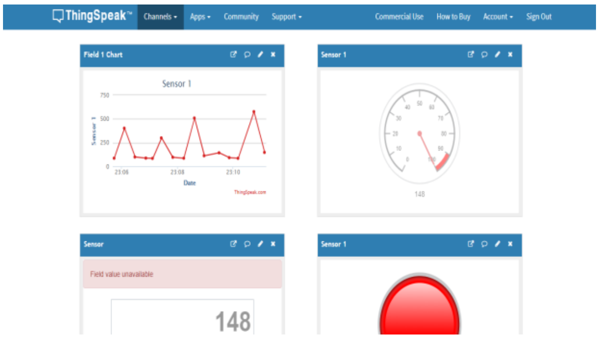


Fig 12. ThingSpeak monitoring system

Figure 12 is a ThingSpeak-based monitoring display design. The monitoring display design monitors the condition of a Landfill Gas (LFG) pipe connection. Monitoring starts with line graph monitoring (top left), then analog meter monitoring (top right), then digital meter monitoring (bottom left), and last is indicator monitoring (bottom right). Furthermore, the displayed data can also store in detailed documents as weekly, monthly, or yearly reports.

4. Result and Discussion

The tool's design is complete and ready to use in this section of the results. The results obtained from this study are an internet of things (IoT) prototype in the form of a Landfill Gas (LFG) pipe leakage monitoring tool based on the NODE MCU ESP8266 microcontroller. Implement the monitoring system/tools for the Gas pipe leak in PLTSa.

4.1. Final Design of the Monitoring Tool

The tool's design is complete and ready to use in this section of the results. The results obtained from this study are an internet of things (IoT) prototype in the form of a Landfill Gas (LFG) pipe leakage monitoring tool based on the NODE MCU ESP8266 microcontroller. Implement the monitoring system/tools for the Gas pipe leak in PLTSa.

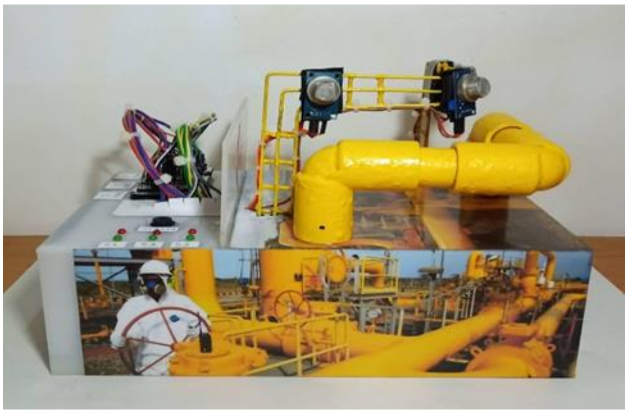


Fig 13. The final design of the monitoring system tools

Figure 13 is the final design of the monitoring system tools with all the other components. The Landfill Gas (LFG) pipe leakage monitoring model divides into two parts, the left and right sides like in the picture. On the left is the Control Room for installing 3 NODE MCU ESP8266, 3 LED indicators (normal and dangerous), three buzzers, and the entire wiring terminal (located inside the control room). On the right is a model of the Landfill Gas (LFG) pipe installation, a frame for placing sensors and installing sensor wiring, as well as 3 MQ-4 sensors themselves as a simulation of 3 sensors laying in 3 different parts so that in the simulation there are three parts to be monitored.

4.2. The Installation of ESP8266 MCU NODE

In the ESP8266 MCU NODE installation, there is an assembling or a combination of 3 ESP8266 MCU Node components. The ESP8266 MCU Node circuit with other components is in Figure 14.

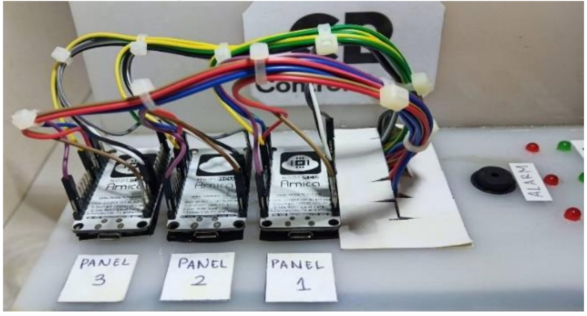


Fig 14. The installation of Node MCU ESP8266 and other components

Figure 14 is The Installation of Node MCU ESP8266. This installation connects the pins on the ESP8266 MCU NODE to the sensor and other components. The number of pins used for the ESP8266 MCU NODE is nine, including 4 GND pins, one 3V3 pin, one analog pin, and three digital pins. The analog pin is an A0 pin, and the three digital pins are the D2 pin, D3 pin, and D4 pin.

4.3. The installation of MQ-4 gas sensor

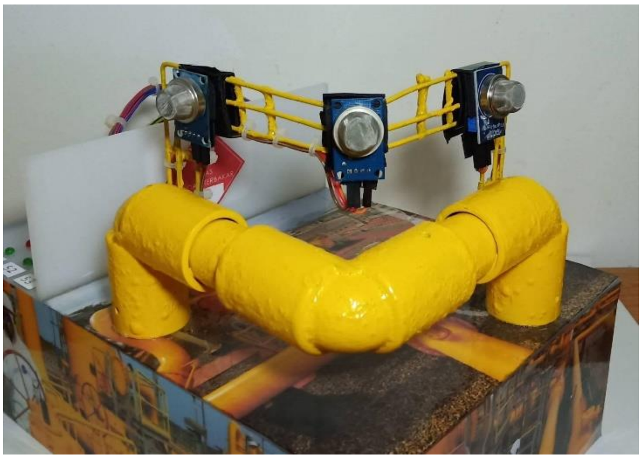


Fig 15. The installation of MQ-4 gas sensor and other components

Figure 15 is an Installation of the MQ-4 Gas Sensor and Other Components. In this installation, there are 3 MQ-4 gas sensors. The MQ-4 gas sensor detects Landfill Gas (LFG) pipeline leaks. Three pins are used on each MQ-4 gas sensor to manufacture this installation. The three pins are The VCC pin, the GND pin, and the Analog Out pin.

4.4. The Installation of LED Indicator and Buzzer

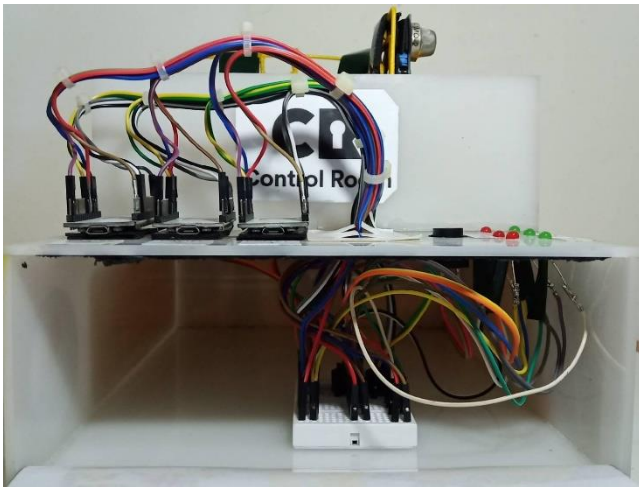


Fig 16. The installation of LED indicator, buzzer, and other components

Figure 16 is a series of LED indicators and buzzers used in this installation to indicate the condition of the Landfill Gas (LFG) pipe and provide an early warning (in the form of an alarm) at the Landfill Gas (LFG) pipe installation location. In "normal" conditions (green LED lights up) and until there is a gas element on the MQ-4 gas sensor, the indication changes to active in "dangerous" conditions (red LED lights up) accompanied by the activation of the buzzer as a warning alarm.

4.5. Black Box Testing

In the tool testing section, use black box testing to test the function of the tools. Table 4 below explains the details of testing using black box testing.

Table 3. Tools testing result

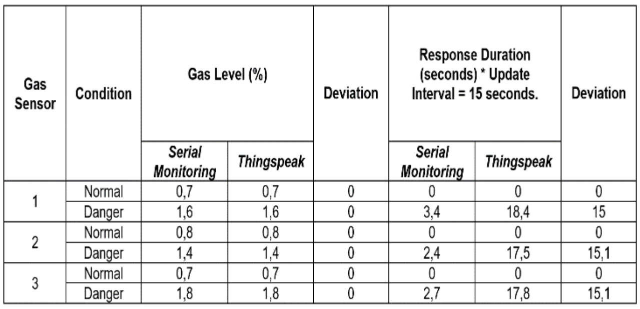
No	Tools Testing	Test Step	Actual Result	Status
1	NODE MCU ESP826	Connect the laptop with NODE MCU ESP8266 using a micro-USB cable	NODE MCU ESP8266 read in Visual Studio Code on a laptop	OK
2	NODE MCU ESP826	Upload the program into the NODE MCU ESP8266 using Visual Studio Code	Upload the program to the NODE MCU ESP8266	OK
3	MQ-4 Gas Sensor	Connect NODE MCU ESP8266 and gas sensor MQ-4. Then upload the program list on NODE MCU with Visual Studio Code application	NODE MCU ESP8266 can read and detect the sensor, then connect it.	OK
4	MQ-4 Gas Sensor	Test the MQ-4 Gas sensor input by providing methane gas	NODE MCU ESP8266 reads or detects the sensor, then detect the value of the sensor on the LED indicator and serial monitoring	OK

			of the Visual Studio Code	
5	LED	Connect the indicator LED to the NODE MCU ESP8266, connect the laptop to the NODE MCU ESP8266 using a micro-USB cable, then upload the indicator LED test program.	The LED can indicate the condition of the Landfill Gas (LFG) pipe installation location.	OK
6	Buzzer	Connect the buzzer to the NODE MCU ESP8266, connect the laptop to the NODE MCU ESP8266 using a micro-USB cable, then upload the buzzer	The buzzer can give a warning alarm only when the red LED is active.	OK
7	Data value adjustment	Connect the ThingSpeak application from the laptop to the NODE MCU ESP8266 via the internet network, and make sure the prototype has all components and power supplies installed. Read data in 2 conditions (with MQ-4 gas sensor triggering by methane gas), such as normal and dangerous conditions.	The data on the visual studio code monitoring serial is the same as those on the ThingSpeak monitoring application.	OK

4.6. The Results of Functional Tool

At the implementation stage, there is the testing of the functionality of the tool. The final design of the monitoring system Tests after complete black box testing and found no errors. Testing of the function of monitoring tools is in three experiments. The table below shows the experimental results of testing the functions of the monitoring tool.

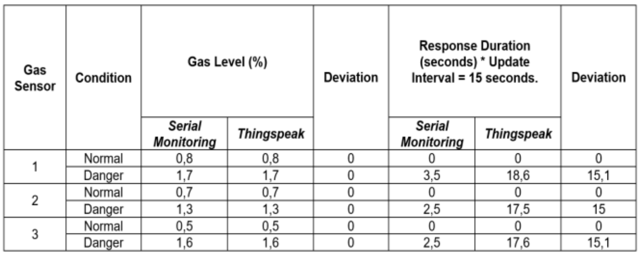
Table 4. The first test of the functional of the tool



Gas Sensor	Condition	Gas Level (%)		Deviation	Response Duration (seconds) * Update Interval = 15 seconds.		Deviation
		Serial Monitoring	*Thingspeak*		*Serial Monitoring*	*Thingspeak*	
1	Normal	0,7	0,7	0	0	0	0
	Danger	1,6	1,6	0	3,4	18,4	15
2	Normal	0,8	0,8	0	0	0	0
	Danger	1,4	1,4	0	2,4	17,5	15,1
3	Normal	0,7	0,7	0	0	0	0
	Danger	1,8	1,8	0	2,7	17,8	15,1

Table 4 is a table of the results of the first implementation regarding the usefulness of the monitoring tool. Table 5 shows a difference between the values of the ThingSpeak database and Serial monitoring. The difference in value is related to gas content and response time duration.

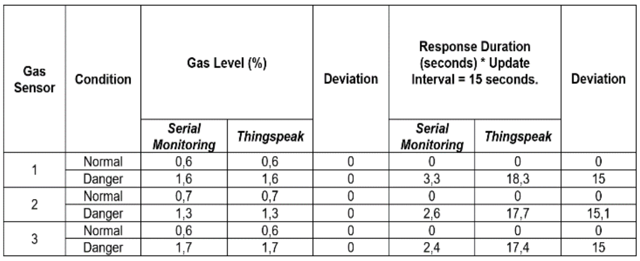
Table 5. The second test of the functional of the tool



Gas Sensor	Condition	Gas Level (%)		Deviation	Response Duration (seconds) * Update Interval = 15 seconds.		Deviation
		Serial Monitoring	*Thingspeak*		*Serial Monitoring*	*Thingspeak*	
1	Normal	0,8	0,8	0	0	0	0
	Danger	1,7	1,7	0	3,5	18,6	15,1
2	Normal	0,7	0,7	0	0	0	0
	Danger	1,3	1,3	0	2,5	17,5	15
3	Normal	0,5	0,5	0	0	0	0
	Danger	1,6	1,6	0	2,5	17,6	15,1

Table 5 is a table of the results of the second implementation regarding the usefulness of the monitoring tool. Table 6 shows a difference between the values of the ThingSpeak database and Serial monitoring. The difference in value is related to gas content and response time duration.

Table 6. The third test of the functional of the tool



Gas Sensor	Condition	Gas Level (%)		Deviation	Response Duration (seconds) * Update Interval = 15 seconds.		Deviation
		Serial Monitoring	*Thingspeak*		*Serial Monitoring*	*Thingspeak*	
1	Normal	0,6	0,6	0	0	0	0
	Danger	1,6	1,6	0	3,3	18,3	15
2	Normal	0,7	0,7	0	0	0	0
	Danger	1,3	1,3	0	2,6	17,7	15,1
3	Normal	0,6	0,6	0	0	0	0
	Danger	1,7	1,7	0	2,4	17,4	15

Table 6 is a table of the results of the third implementation regarding the usefulness of the monitoring tool. Table 7 shows a difference between the values of the ThingSpeak database and Serial monitoring. The difference in value is related to gas content and response time duration.

4.7. Test Research Implication

The research implication is the calculation of the relative error or error rate. In research implications, the symbols unit of gas content is percent (%). Meanwhile, the response duration is symbolized by seconds (s). Based on the result data from the implementation of the three previous tool tests, the calculation of the relative error or error rate is as follows:

1. Sensor 1 (Normal condition)

Deviation in Gas Content Value = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

Deviation in response duration = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

2. Sensor 1 (Danger Condition)

Deviation in Gas Content Value = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

Deviation in response duration = 15 + 15.1 + 15

= (45.1/3) – 15

Error rate = 0.03

Percentage = 0.03 x 100%

= 3%

3. Sensor 2 (Normal condition)

Deviation in Gas Content Value = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

Deviation in response duration = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

4. Sensor 2 (Danger condition)

Deviation in Gas Content Value = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

Deviation in response duration = 15.1 + 15 + 15.1

= (45.2/3) – 15

Error rate = 0.06

Percentage = 0.06 x 100%

= 6%

5. Sensor 3 (Normal condition)

Deviation in Gas Content Value = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

Deviation in response duration = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

6. Sensor 3 (Danger condition)

Deviation in Gas Content Value = 0 + 0 + 0

= 0 / 3

Error rate = 0

Percentage = 0 x 100%

= 0%

Deviation in response duration = 15.1 + 15.1 + 15

= (45.2/3) – 15

Error rate = 0.06

Percentage = 0.06 x 100%

= 6%

5. Conclusion

The prototype design of the Landfill gas (LFG) pipe leakage monitoring tool is to use the main components in the form of a microcontroller and NODE MCU ESP8266. The MQ-4 gas sensor, LED, Buzzer, and cable are other supporting components. The gas leak detector uses the MQ-4 sensor. The LED is an indication of the condition of the gas pipeline installation. A buzzer as a gas leak alarm. The ESP8266 MCU NODE, the main component, is a controller and an internet of things (IoT) based data sender. With the MCU Node, it can send data about the input to the sensor with the help of the ThingSpeak application. The use of ThingSpeak, IoT, and Node MCU applications aims to be able to monitor gas pipelines remotely. In the tool's design, remote tools can work in real time with an update interval of 15 seconds. The error rate test is three times testing. The error rate testing experiment for three trials was on 3 MQ-4 gas sensors. The experimental results are the percentage of gas levels against serial monitoring and ThingSpeak in normal and dangerous conditions. Under normal circumstances, the error rate on the sensor is 0%, sensor 2 is 0%, and sensor 3 is 0%. Furthermore, testing the MQ-4 gas sensor against a state of danger, the error rate on sensor 1 is 0%, sensor 2 is 0%, and sensor 3 is 0%.

Tests on the duration of the gas response to serial monitoring and ThingSpeak are under normal and dangerous conditions. In normal testing, the error rate on sensor 1 is 0%, sensor 2 is 0% and sensor 3 is 0%. Furthermore, in a state of danger, the error rate on sensor 1 is 3%, sensor 2 is 6%, and sensor 3 is 6%. Based on the test data, the system has a relative error (Error Rate) at a response duration of 3% to 6%. While the gas content value has a relative error (Error Rate) of 0%. 5.2 Suggestions 1. This system still needs development in terms of the Thingspeak display so that it can be made with a one-display monitoring system for many monitoring points and in terms of the Thingspeak update interval, which is only capable of updating monitoring at the fastest of 15 seconds per data. 2. In terms of application in the actual Landfill Gas (LFG) pipe installation, it is necessary to pay attention to the distance from the gas sensor to the controller so that there are no delays or delays in sending data because the cable is too long.

Author contributions

Abdul Aziz: Conceptualization, Methodology, Software, Field study Data curation, Writing-Original draft preparation, Software, **Amalia Zahra:** Validation., Field study, Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] A. T. Bakesheva, V. G. Fetisov, and V. V. Pshenin, “A refined algorithm for leak location in gas pipelines with determination of quantitative parameters,” *Int. J. Eng. Res. Technol.*, vol. 12, no. 12, pp. 2867–2869, 2019.

[2] P. Dadkani, E. Noorzai, A. Ghanbari, and A. Gharib, “Risk analysis of gas leakage in gas pressure reduction station and its consequences: A case study for Zahedan,” *Heliyon*, vol. 7, no. 5, p. e06911, May 2021, doi: 10.1016/j.heliyon.2021.e06911.

[3] M. M. Rezinkina, O. L. Rezinkin, K. L. Chrzan, S. A. Lytvynenko, and N. V. Veselova, “Simulation of the Power Transmission Lines Electrical Field to Ensure Safe Navigation of the Unmanned Aerial Vehicles at Their Monitoring,” in *2019 Modern Electric Power Systems (MEPS)*, Sep. 2019, pp. 1–4, doi: 10.1109/MEPS46793.2019.9394999.

[4] F. R. Doni and A. M. Lukman, “Implementasi Monitoring Kamera CCTV Online Dengan Penerapan Hik-Connet,” *EVOLUSI J. Sains dan Manaj.*, vol. 9, no. 1, pp. 12–21, Mar. 2021, doi: 10.31294/evolusi.v9i1.9984.

[5] N. V. S. Korlapati, F. Khan, Q. Noor, S. Mirza, and S. Vaddiraju, “Review and analysis of pipeline leak detection methods,” *J. Pipeline Sci. Eng.*, vol. 2, no. 4, p. 100074, Dec. 2022, doi: 10.1016/j.jpse.2022.100074.

[6] Y. Hendrian and D. Dalhar, “Sistem Monitoring Local Area Network Pelanggan PT. PHE ONWJ Pada PT. Patrakom Jakarta,” *J. Tek. Komput.*, vol. 7, no. 1, pp. 55–63, Feb. 2021, doi: 10.31294/jtk.v7i1.9217.

[7] N. H. L. Dewi, M. F. Rohmah, and S. Zahara, “PROTOTYPE SMART HOME DENGAN NODEMCU ESP8266 BERBASIS IOT,” *J. Ilm. Tek.*, vol. 1, no. 2, pp. 101–107, May 2022, doi: 10.56127/juit.v1i2.169.

[8] D. G. K. Yoga, I. M. A. D. Suarjaya, and I. P. A. E. Pratama, “The Development of Prototype Data Delivery System Based on LoRa and Mesh Topology,” *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 10, no. 1, pp. 46–58, 2022.

[9] F. Romisa, “DETEKSI PINEAP PADA FIRMWARE WIFI PINEAPPLE MENGGUNAKAN SMARTPHONE BERBASIS ANDROID,” *Ilk. J. Ilm.*, vol. 8, no. 3, pp. 204–211, Dec. 2016, doi: 10.33096/ilkom.v8i3.68.204-211.

[10] R. Syam, *Dasar-Dasar Teknik Sensor*. Makassar: Fakultas Teknik, 2013.

[11] D. Muthmainnah, N. K. Suryati, and Y. S. Mulyani, “Upaya Peningkatan Nilai Produk Ikan Sidat Melalui Partisipasi Masyarakat,” *Pros. Semin. Nas. Lahan Suboptimal ke-8 Komod. Sumber Pangan untuk Meningkat. Kualitas Kesehat. di Era Pandemi Covid-19*, pp. 237–244, 2020.

[12] N. Wahyuni, S. Syaifurrahman, and J. Islami, “Instalasi PLTS Skala Rumah Tangga dengan Lampu Led Dc Hemat Energi bagi Masyarakat Terpencil di Kabupaten Kubu Raya, Kalimantan Barat,” *J-ABDIPAMAS (Jurnal Pengabdi. Kpd. Masyarakat)*, vol. 3, no. 2, p. 17, Oct. 2019, doi: 10.30734/j-abdipamas.v3i2.570.

[13] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban, “LED Based Indoor Visible Light Communications: State of the Art,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 3, pp. 1649–1678, 2015, doi: 10.1109/COMST.2015.2417576.

[14] D. Mardiati, A. Mering, and D. Miranda, “HUBUNGAN ANTARA KEPERCAYAAN DIRI DENGAN MOTIVASI BELAJAR PADA ANAK KELOMPOK B DI TK Desi,” *J. Pendidik. dan Pembelajaran Khatulistiwa*, vol. 4, no. 9, pp. 1–11, 2015.

[15] I. Iksal, S. Sumiati, and H. Harizal, “Rancang Bangun Prototype Penanganan Dini Dan Pendeteksi Kebocoran Lpg Berbasis Mikrokontroler Melalui Sms,” *J. PROSISKO*, vol. 3, no. 2, pp. 26–32, 2016.

[16] A. Rohmanu and D. Widiyanto, “SISTEM SENSOR JARAK AMAN PADA MOBIL BERBASIS MIKROKONTROLLER ARDUINO ATMEGA328,” *Sist. Sens. Jarak Aman Pada Mob. Berbas. Mikrokontroller Arduino Atmega328*, vol. 3, no. 1, p. 9, 2018.

[17] T. Lamberts, G. Fedoseev, M. C. Van Hemert, D. Qasim, H. Linnartz, and K. Chuang, “Methane Formation in Cold Regions from Carbon Atoms and Molecular Hydrogen,” *Astrophys. J.*, vol. 928, no. 1, p. 48, 2022, doi: 10.3847/1538-4357/ac51d1.

[18] R. Puspitasari, M. Muladno, A. Atabany, and S. Salundik, “Produksi Gas Metana (CH4) dari Feses Sapi FH Laktasi dengan Pakan Rumput Gajah dan Jerami Padi,” *J. Ilmu Produksi dan Teknol. Has. Peternak.*, vol. 3, no. 1, pp. 40–45, Jan. 2015, doi: 10.29244/3.1.40-45.

[19] R. S. M. Permana, L. Puspitasari, and S. S. Indriani, “Industri film Indonesia dalam perspektif sineas Komunitas Film Sumatera Utara,” *ProTVF*, vol. 3, no. 2, p. 185, Sep. 2019, doi: 10.24198/ptvf.v3i2.23667.

[20] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, “Pengujian Black Box Testing pada Aplikasi Action &amp; Strategy Berbasis Android dengan Teknologi Phonegap,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, Dec. 2018, doi: 10.30998/string.v3i2.3048.

[21] D. Saputra, “Perbandingan Teknologi System Software, Application Software, Embedded Software Dan Web Applications,” *J. Bangkit Indones.*, vol. 5, no. 1, p. 74, Mar. 2016, doi: 10.52771/bangkitindonesia.v5i1.67.

[22] Y. D. Wijaya and M. W. Astuti, “PENGUJIAN BLACKBOX SISTEM INFORMASI PENILAIAN KINERJA KARYAWAN PT INKA (PERSERO) BERBASIS EQUIVALENCE PARTITIONS,” *J. Digit. Teknol. Inf.*, vol. 4, no. 1, p. 22, Mar. 2021, doi: 10.32502/digital.v4i1.3163.

[23] W. E. Perry, *Effective Methods for Software Testing*, Third. Canada: Wiley Publishing Inc., 2006.

[24] N. Putri, N. Agung Prabowo, and R. A. Widyanto, “Implementasi Metode Prototyping pada Perancangan Aplikasi Electronic Ticket (E-Ticket) berbasis Android,” *J. Komtika (Komputasi dan Inform.*, vol. 3, no. 2, pp. 62–68, Apr. 2020, doi: 10.31603/komtika.v3i2.3474.

[25] E. W. Fridayanthie, H. Haryanto, and T. Tsabitah, “Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web,” *Paradig. - J. Komput. dan Inform.*, vol. 23, no. 2, Sep. 2021, doi: 10.31294/p.v23i2.10998.