Development of an IoT Cloud Control System for Managing Pressurized Gas Containers

FRANCESCO ZITO, NICOLA IVAN GIANNOCCARO, ROBERTO SERIO,
SERGIO STRAZZELLA
Department of Engineering Innovation,
University of Salento,
Via per Monteroni 73100 Lecce,
ITALY

Abstract: - Automatic remote management of gas cylinders is an important task for gas companies, especially if it saves time and money on replacement and maintenance. The system presented here can be used in various industries that require monitoring of cylinders containing gas, such as industrial, medical, and food. The main advantage offered is the ability to arrange for cylinder replacement in advance to speed up, minimize downtime, and facilitate logistics. Monitoring uses a microcontroller from the Arduino family, which can continuously detect the state inside the cylinder via a pressure transducer. The customer can monitor the system by viewing the cylinder's status locally or through an online dashboard. In addition, if the level inside the cylinder becomes critical, the system turns on a warning LED light and sends an Alert message to the Cloud containing location information. The alert message was generated in the AWS IoT Cloud environment using the MQTT protocol and will be received on a mobile device owned by the cylinder replacement operator. Finally, an energy analysis has been carried out to evaluate the autonomy characteristics of the device.

Key-Words: - Technical gas cylinder; Mkr Wi-Fi 1010 microcontroller; pressure detection; IoT, energy design, AWS.

Received: April 26, 2022. Revised: August 23, 2023. Accepted: November 19, 2023. Published: December 31, 2023.

1 Introduction

One of the main problems encountered in a production cycle is waiting times, which cause stops to the detriment of the production itself. Avoiding some situations that could cause interruptions is the common thread, [1] that led to the development of this research. There are many real industrial situations in which the gas cylinder becomes empty and operators must stop operating while waiting to replace the cylinder. Just think, for example, of an operator using an argon cylinder in a TIG (Tungsten Inert Gas) welding operation, [2] or the use of food gases (carbon dioxide, nitrogen, argon, and oxygen) in the food and beverage industry, [3] and many others.

This study investigated possibilities for reducing process interruptions through automatic and remote sensing of gas cylinder pressure and its precise location. The primary objective is to develop a digital pressure manometer capable of detecting the pressure inside cylinders. Unlike traditional pressure manometers, which merely display pressure readings, this digital solution eliminates the need for the constant presence of the operator near the cylinder to ensure an uninterrupted gas supply.

The scope of application of the device developed in this study is broad, as it can be integrated into various situations requiring gas pressure monitoring, whether used in a production cycle or stored in depots. The system ensures continuous cylinder monitoring and provides real-time alerts when the cylinder is nearly empty, facilitating preventive replacement.

The model created is a point of reference to be replicated for any control system that requires monitoring an industrial parameter; with low investment, it is possible to make hardware components not born for Industry 4.0 suitable for the IoT environment.

The proposed control system was developed using a low-cost microcontroller of the Arduino family, [4], which can continuously detect internal pressure by communicating with a suitable pressure sensor. Similar recent IoT applications for monitoring gas cylinders are described in, [5], [6], [7], [8], [9], [10], [11], [12], with different types of sensors and purposes but the same MCU.

In our case, the realized IoT application adds to the measure of gas pressure inside the cylinder and the localization. Recent research using Arduino for GPS using a GSM module is shown in, [13].

In the proposed application, all the data collected are displayed locally through a simple LCD screen and collected remotely in a database. The system has been integrated with a GPS module, [14], that geolocates the cylinder and a GMS module, [15] equipped with a SIM to send the alert message. The pressure manometer is located near the cylinder outlet pipe. It is specially designed to make it possible to display the pressure value and to visualize an alert consisting of the lighting of a red LED once the minimum threshold has been reached. Remotely, the person in charge of replacing the cylinder must have a device that allows the reception and display of the alert message, whether a simple smartphone or a mobile device.

The system can be customized in several aspects, such as feedback times and the pressure or battery level at which the alarm will be sent. The operator can remotely control the cylinders via a graphical interface by integrating the proposed control system. Geolocation plays an important role, being able to easily identify the position of the empty cylinder in the set of several hundred cylinders, speeding up their replacement. Finally, an energy analysis has been carried out to evaluate the autonomy characteristics of the proposed device, demonstrating its applicability for industrial purposes.

2 Hardware and Firmware

The prototype device consists of two parts:

•The first part is a box with the microcontroller, GPS module, and batteries to power the system. The box has an LCD screen that shows the data revealed by the pressure sensor and the status of the batteries in real time. The box is attached to the cylinder via a Velcro strap. This docking system was chosen because it allows the box to be assembled and disassembled very quickly, thus facilitating logistical operations for cylinder refilling or maintenance.

•The second part is a hydraulic pipe used for pressure measurement. The pipe has two fittings for the analog pressure sensor and a differential pressure gauge. The pipe nozzle is installed upstream of the rolling valve and downstream of the cylinder shut-off valve. The system layout is shown in Figure 1.

The main hardware components (shown in Figure 2) are:

▪MKR Wi-Fi 1010 microcontroller, which, by integrating the ESP 32 module, allows connection to a Wi-Fi network (detailed in 2.1),

▪NEO 6m V2 GPS module (detailed in 2.2), which allows the cylinder to be located,

▪Analog Pressure transducer (detailed in 2.3) for pressure measurement,

▪LCD and a red LED for direct visualization of the pressure value.

2.1 MKR Wi-Fi 1010

The MKR Wi-Fi 1010 (Figure 3) is a miniature-sized module containing a SAMD21G18A Processor, the Nina W102 Module, and a crypto chip (the ATECC508). It has been designed to speed up and simplify the prototyping of IoT applications based on Wi-Fi connectivity, thanks to the U-BLOX NINA-W10 ESP32 module's flexibility and low power consumption.

The Arduino MKR WIFI 1010 can be powered using external 3.7 Volt or 5 Volt Li-Po batteries, recharging the Li-Po battery while running on external power, [16].

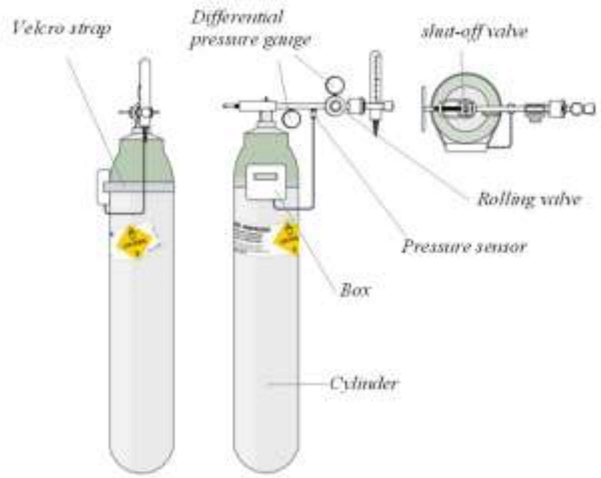

Fig. 1: System Layout

The microcontroller's logic locally monitors the cylinder level and sends the appropriate feedback, as shown in Figure 4.

The system's main task is to send an alert message that the operator responsible for replacing the cylinder will receive. In this way, the criticality linked to the operational stop required for the replacement is overcome. This will allow it to integrate seamlessly into the production cycle, improving it.

The firmware is structured as follows:

At boot, the GPS and pressure sensors are initialized; a microcontroller pin is also initialized as an analog input. This pin is physically connected to a voltage divider that allows the state of charge of the batteries to be derived; this information will be added to the message sent.

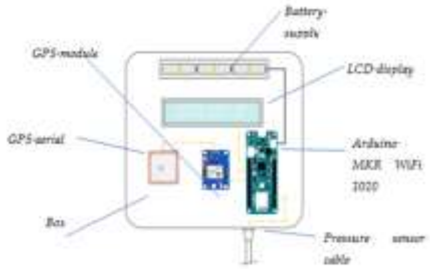

Fig. 2: Hardware components in the box

Fig. 3: MKR Wi-Fi 1010

The control cycle occurs every two minutes, mainly based on checking the pressure level. If the pressure is low, the microcontroller connects to the MQTT, [17], broker via Wi-Fi and sends an alert message. On the other hand, if the state of charge of the cylinders is above the threshold, the microcontroller sends feedback every hour.

The message sent contains the following information:

• state of charge of the batteries,
• state of charge of the cylinders,
• GPS coordinates.
• System ID

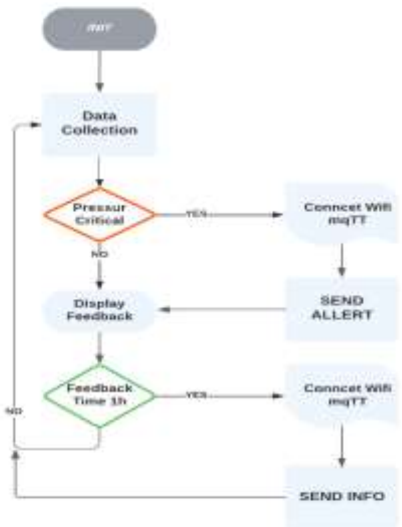

Fig. 4: Firmware logic flow chart

2.2 GPS Module

The integration of the GPS module allows the localization of cylinders. It is useful to speed up replacement operations, and it can be used to optimize logistics operations.

Specifically, the GY-NEO 6M V2 GPS module was selected (Figure 5). The main part of the module is a NEO-6M GPS chip from U-Blox 6. It can track up to 22 satellites across 50 channels and achieves a sensitivity level of -161dB tracking while consuming only 45mA of supply current, [18]. Power-saving mode (PSM) reduces system power consumption by selectively turning on and off certain parts of the receiver. This significantly reduces the electric current of the module to only 11 mA, making it compatible for use in stand-alone applications.

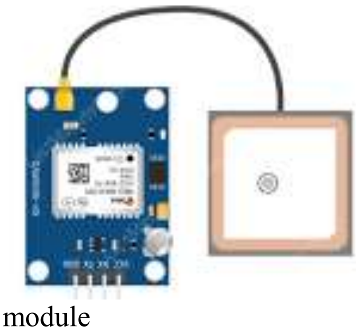

Fig. 5: GPS module

2.2.1 GPS Module Operation

The LED flashing on the NEO-6M GPS module indicates the Position Fix status. It will flash at various speeds depending on the state it is in:

• No flash: it means it is searching for satellites.
• Flashes every second: means that the position fix has been found.

The operating voltage of the NEO-6M chip is 2.7 to 3.6 V. However, the module comes with MICREL's MIC5205 ultra-low dropout 3V3 regulator. The logic pins are also 5 volt tolerant, so you can easily connect to an Arduino or any 5V logic microcontroller without any logic-level converters.

It is possible to read data formatted according to the NMEA 0183 standard by connecting the pins of the GPS module to Arduino using UASRT NMEA 0183, a combined electrical and data specification for communication between marine electronics. NMEA 0183 is a proprietary protocol issued by the National Marine Electronics Association for use in boat navigation and control systems, e.g., the standard is used for the operation of many instruments such as depth sounders, sonar, anemometers, gyrocompasses, and GPS receivers,

[19], [20]. All NMEA's sentences have a structure like:

```
$PREFIX, data1, data2.. dataN-1,dataN*CHECKSUM
```

The prefix is the first part of the string, which is used to specify what type of talker is; in this case, the device is a GPS module, so the prefix is GP followed by the type of the sentence. All sentences are identified with three letters, e.g., RMC, RMB, etc. Specifically, the information provided by the GPGGA sentence is used for the extrapolation of the Global Positioning System fixed data as reported below:

```
$GPGGA: hhmmss.ss, llll.ll, a, yyyyy.yy, a, x, xx, xx, xx, M, xx, M, xx, xxxx
```

2.3 Pressure Transducer

The selected commercial pressure transducer (Sensata, BTE7100 series) has three main characteristics:

• It's powered by a voltage of 5 volts corresponding to the output voltage supplied by the used microcontroller.
• Its measurement range covers pressure detection from up to 200 bars.
• The electrical output signal is in [0-5] V range.

The purpose is to have a general indication of the charged state of the cylinder; after experimental analysis, it was derived that commercial sensors (sensitivity in the order of 1% of full scale) are suitable for the purpose. The pressure transducer must be calibrated to work properly. The sensor was calibrated to a compressed air cylinder via a connecting pipe and a manual manometer for the calibration values (Figure 6).

One hundred fifty tests were performed by gradually increasing the pressure from 0 bar to 5 bar for the pressure transducer calibration. The sensor was cabled to Arduino MKR Wi-Fi 1010 for data analysis, and the calibration curve was obtained with test results. The equation that best interpolates the data trend was derived from experimental data in Table 1.

Fig. 6: Setup for the pressure sensor calibration.

Table 1. Pressure data

Bar	Analog Readings Average	Number Of Measures	Analog Readings Standard Deviation
0,0	562,5	54	7,8
1,4	1140,7	13	57,9
1,5	1178,0	13	84,5
1,8	1427,4	5	11,4
2,0	1563,6	5	14,5
2,4	1653,0	5	10,9
2,5	1789,4	5	10,7
2,8	1940,6	5	12,5
3,0	2060,0	5	2,8
3,4	2194,5	6	17,4
3,4	2215,9	7	18,4
3,5	2309,5	13	39,2
4,4	2751,0	7	4,7
4,5	2805,0	7	4,6

2.3.1 Pressure Linear Regression

The chart shows (Figure 7) that the coefficient of determination R^2 value tends toward unity, so linearity can be assumed between the value read by the microcontroller and the cylinder pressure. The coefficients of the straight line were obtained by generating a linear regression model from the experimental data obtained.

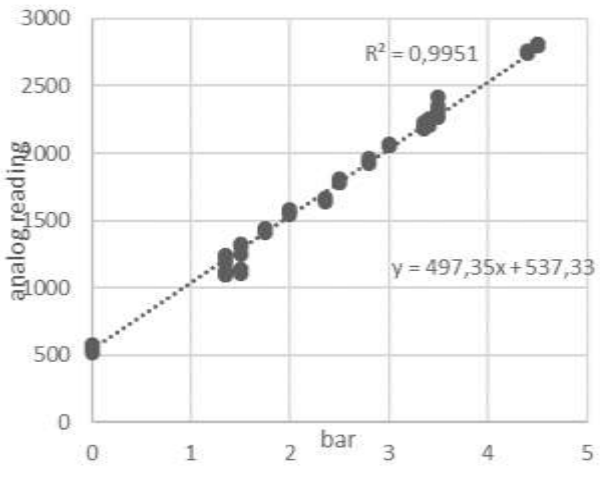

Fig. 7: Interpolation curve obtained by the calibration

The analysis results in Table 2 show the standard error and R squared value on the linear regression carried out by 150 measurements and coefficients of the linear regression line. Figure 8, Figure 9 and Figure 10 show the distribution of residuals from the statistical analysis, the graph of approximations, and the normal probability graph.

Table 2. Linear regression is outgoing

Regression statistics	
R squared	0,9951
Standard error	54,83
Angular coefficient	497,35
Intercept	537,33
Measurements	150

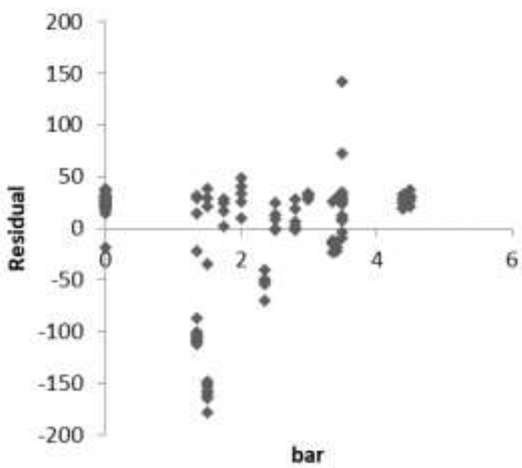

Fig. 8: Tracing of residuals

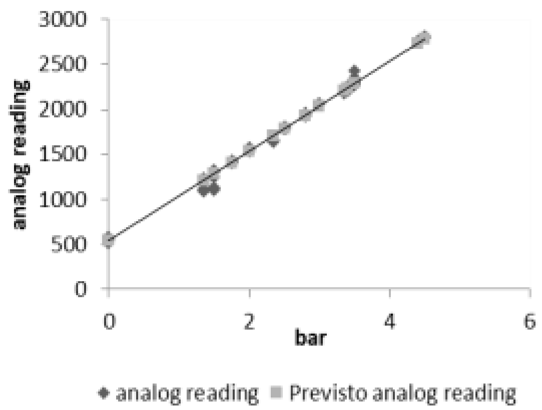

Fig. 9: Linear approximation

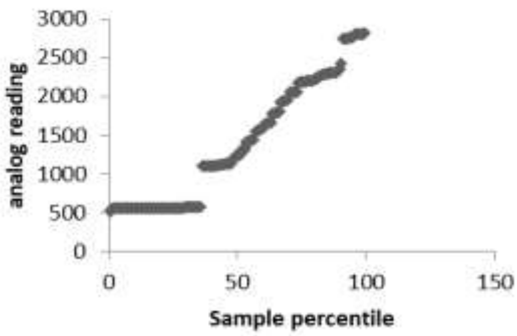

Fig. 10: Tracing of normal probability

3 AWS IoT

The proposed system is based on the AWS IoT core for allowing a smart use of the data. This service easily allows the connection between a physical device and the Cloud. AWS IoT Core can support billions of devices and trillions of messages; it can reliably process and route those messages securely from AWS endpoints to other devices. The logical part connection is schematized in Figure 11. Devices can connect to AWS IoT Core using the following protocols: HTTP, Web-Socket, and MqTT.Message Queue Telemetry Transport (MQTT), [17], is an ISO standard lightweight messaging protocol that sits on top of TCP/IP. It was designed for situations where low impact is required and where bandwidth is limited. The communication scheme is published-subscribe and requires a message broker to work.

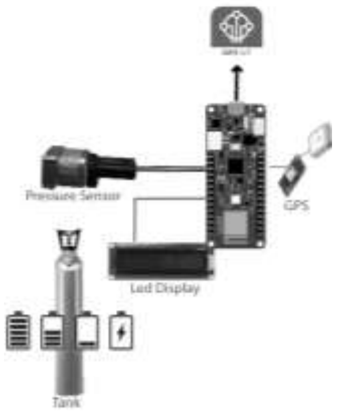

Fig. 11: Design system layout

Figure 12 was extracted from the MQTT tester made available in the AWS IoT core service. The image shows the structure of the JSON message underlying the communication between the system and the Cloud.

▼ arduino/outgoing

{
"SystemID": 001,
"Volt[v]": 11,
"pressure[bar]": 4,
"state": "Fully charge",
"latitude[N]": 40.33449633,
"longitude[E]": 18.06630367
}

Fig. 12: JSON payload format

3.1 Security on AWS IoT Core

AWS IoT Core requires devices that connect using the MQTT protocol to use X.509 certificates for authentication, [21]. A CSR (Certificate Signing Request) must be generated to do that. This will allow the microcontroller to be associated with an AWS' thing,' allowing a unique connection to its physical counterpart. Another relevant aspect is the definition of the policy of 'things.' It allows establishing and limiting the actions that individual 'things' can undergo and perform.

3.2 Database Creation

Therefore, it was decided to support the developed control system with a non-relational database as it is the most flexible and simplest solution to implement. A database was then created with a partition key (cylinder ID) and a sort of key (time), [22]. Thanks to the IoT rule, it is possible to check the JSON content of MQTT messages arriving at the relevant topic. If the partition key is present (and satisfies the necessary parameters), it is possible to transfer the message content or part of it to the database. The micro-service will organize the table with the necessary column division and add the timestamps.

4 Power and Energy Analysis

Preliminary experimental tests show that the average power consumption is 0.5 W.
Assuming constant power, the hourly consumption equals 0.5 [Wh/h]. The application must be guaranteed an energy autonomy of at least two days. Therefore, the energy is expressed in (1 and 2):

$$energy = 0.5 \left[\frac{Wh}{h}\right] \cdot 24 \left[\frac{h}{day}\right] \cdot 2[day] \quad (1)$$

$$energy = 24 \ [Wh] \quad (2)$$

Assuming a 12V battery, its capacity is calculated for conversion from Wh to Ah. (3):

$$BC[Ah] = \frac{energy\ [Wh]}{V[V]} = 2\ [Ah] \quad (3)$$

Such a battery would make the device autonomous for a period sufficient for transport and replacement operations. There are some improving solutions for increasing the prototype autonomy, such as:
- The system has a photovoltaic power generator and a smaller storage system.
- Change the microcontroller using one custom that allows it to enter low power mode by lowering its power consumption.
- Provide an external power supply.

4.1 Battery Charge Measurement

To calculate the battery's charge state, the voltage divider rule was used by taking an analog voltage reading between nodes A and B of the circuit shown in Figure 13, [9], [23], [24].

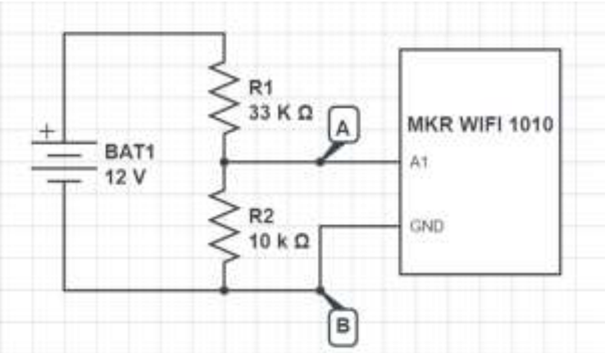

Fig. 13: Voltage divider electrical circuit.

The choice of resistors is related to the maximum voltage readable by the microcontroller of 3.3 volts and the electric current flowing through them, as it affects the system's power consumption, which must be minimized. The choice of resistors is related to the maximum voltage readable by the microcontroller of 3.3 volts and the electric current flowing through them, as it affects the system's power consumption, which must be minimized. Eq. (4 and 5) allow to derive the voltage between A and B (VAB) based on the input voltage (Vin) and the resistors (R1 and R2):

$$V_{AB} = V_{in} * \frac{R2}{R2 + R1} \quad (4)$$

With $V_{in} = 12\,[V]\;\; R1 = 33\,[k\Omega]\;\; R2 = 10\,[k\Omega]$

$$V_{AB} = 12\,V * \frac{10.000\,\Omega}{10.000\Omega + 33.000\,\Omega} \cong 2{,}79\,V \quad (5)$$

4.1.1 Electric Test Circuit

Fig. 14: Electric test circuit

The test circuit described above (Figure 13) was made with the experimental set-up shown in Figure 14, which includes a multimeter (1) connected to the AB node, two resistors (2), an Arduino MKR Wi-Fi 1010 (3), a variable voltage generator (4) and the LCD (5).

4.1.2 Initial Calibration

The code shown in Figure 15 was used to extrapolate the voltage data.

The variable val is used as the AB voltage. It is scaled to compensate for the resampling of the microcontroller, as reported in Eq. (6).

$$val_{scaled} = val * 3{,}3/1024 \quad (6)$$

The values valscaled were recorded for different Vin voltage levels; during the measurements, the actual voltage of node AB was also continuously checked through the multimeter to verify that the circuit was working properly. Table 3 and Figure 12 show a grouped view of the collected data.

```
#include <Wire.h>
#include <hd44780.h>
#include <hd44780ioClass/hd44780_I2Cexp.h>

hd44780_I2Cexp lcd;
const int LCD_COLS = 16;
const int LCD_ROWS = 2;
int analogPin = A3;
float val = 0;
float val_scaled = 0;

void setup()
{
delay(5000);
Serial.begin(9600);
        int status;
        status = lcd.begin(LCD_COLS, LCD_ROWS);
if(status) hd44780::fatalError(status);
lcd.lineWrap();
Serial.println(" reading scaled_reading");
}
void loop()
{
        val = analogRead(analogPin);
        lcd.clear();
lcd.print("reading : ");
lcd.print(val);
delay(2000);
lcd.clear();
Serial.print(" ");
Serial.print(val);
Serial.print(" ");
val_scaled = val*3.30/1024;
lcd.clear();
lcd.print("scaled_reading : ");
Serial.println(val_scaled);
lcd.print(val_scaled);
delay(2000);
lcd.clear();
}
```

Fig. 15: Arduino code for battery charge measurement

Table 3. Data table

V_{in}	Number of measures	V_{AB} real	Average val_scaled
14,50	46	3,37	3,30
13,50	30	3,14	3,15
12,50	13	2,90	2,92
11,50	13	2,67	2,69
10,50	12	2,44	2,46
9,50	14	2,21	2,22
8,50	10	1,98	1,98
7,50	9	1,74	1,76
6,50	11	1,51	1,52

4.1.3 Linear Regression

The coefficients of the straight line were obtained by generating a linear regression model from the experimental data. The analysis results are shown in Table 4 and Figure 16. Table 4 shows the standard error and R squared value on the linear regression carried out by 274 measurements and the coefficient of the linear regression line. Figure 17 shows the graph of the residuals from the statistical analysis.

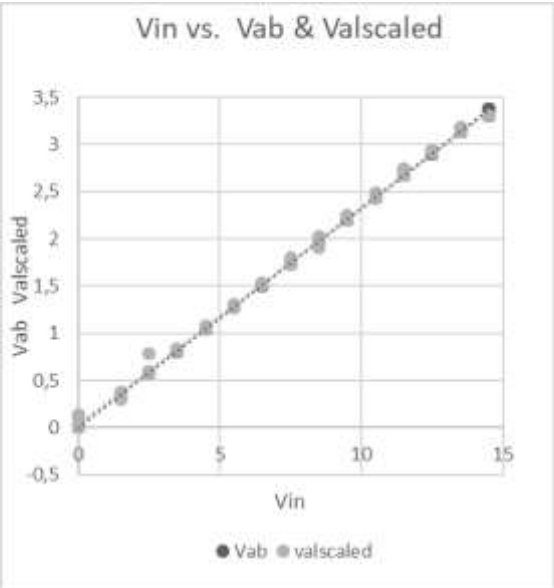

Fig. 16: Data graph

Table 4. Summary output

Regression statistics	
R squared	0,9997
Standard error	0,0403
Measurements	274
Coefficient	0,231

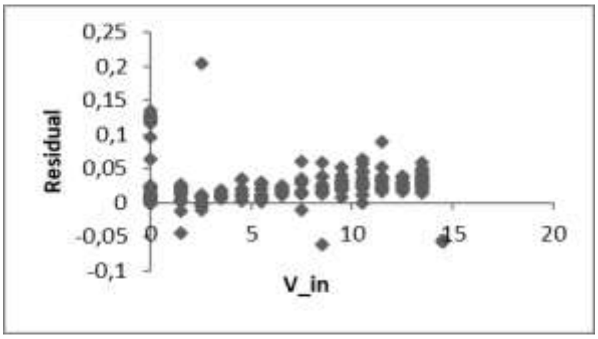

Fig. 17: Tracing of residuals.

Approximating to the second decimal place, the theoretical equation derived from the voltage partition rule is found, so it was possible to confirm Eq. (7) that binds the voltages:

$$V_{AB} = V_{in} * 0{,}23 \quad (7)$$

4.2 Battery Life Evaluation

AWS, [25], services were used to develop a cloud architecture that allows data to be historicized in a database and visualized on a dashboard implemented on Grafana, [26]. This made it possible to visualize battery voltage trends over time to conduct a system energy design review. The graph in Figure 18 has the time reported in the day-hour format in the abscissa and is representative of a system battery discharge cycle. The value is 13 volts when the battery is powered and fully charged, while it decreases by about 0.5 volts when disconnected from the power supply; the tests show that the resulting battery discharge time is about 48 hours, thus validating the energy design.

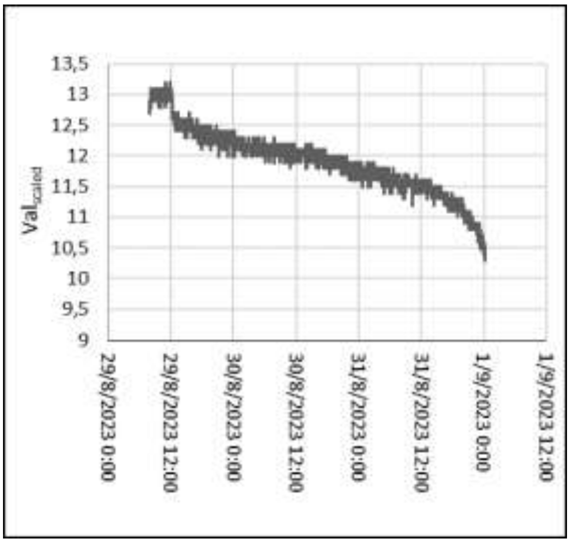

Fig. 18: Timing trend from 29/08/2023 to 01/09/2023 of battery voltage from Grafana Dashboard.

5 Discussion and Potential Future Implementations

The prototype control system can now be considered complete: all the major components required for its implementation have been opportunely connected, and several operational procedures have been introduced for using the AWS IoT cloud-centric environment by taking advantage of the capabilities offered.

For the proposed prototype, two aspects would need to be improved: connectivity and power supply.

The main limitation of the connection is due to the use of Wi-Fi that does not allow communication over long distances (and the need to associate the smart box with the client's Wi-Fi).

There are several possibilities to overcome this problem, such as using radio communications with a gateway (Zigbee, Bluetooth, LoRa) or an LTE 4G SIM card Wi-Fi gateway to be added to the box. The choice of how to proceed will depend on the client's specific case. Regarding the power supply issue, one improvement would be to include a power supply. Still, in this case, the applications of the system should be limited to cases where a power supply is available.

The research found that the system is an excellent tool for optimizing resource management processes in the cylinder rental industry over a wide territory. In addition, a key step for further improvement is integrating artificial intelligence (AI) systems to address emerging challenges and improve the efficiency of management, logistics, and maintenance operations.

5.1 Possibilities of Artificial Intelligence Implementation

5.1.1 Application of Expert Systems

Artificial intelligence-based systems can be used to improve delivery planning and optimize inventory management. These systems can predict peak demand and plan deliveries by analyzing historical data on cylinder consumption, weather patterns, and other factors. In this way, operating costs can be reduced, and customers can be assured of more timely and efficient service: a neural network model could learn from real-time data, adapting dynamically to unexpected changes in demand or traffic conditions. This would enable faster response to sudden changes in customer needs and environmental conditions, [27], [28].

5.1.2 Example of Implementation in a Cylinder Rental Company

Consider, for example, a cylinder rental company that operates over a large territory. A centralized system that constantly monitors demand, cylinder locations, and traffic conditions could be implemented for this company. If a sudden increase in demand occurs in a specific area, the system could automatically reallocate resources, adjust delivery routes, and maintain optimal inventory in each area. In addition, a user interface could be developed through the collected data to allow customers to anticipate their needs and book deliveries in advance, further contributing to resource planning.

5.2 Realized Benefits and Practical Impact

The Artificial Intelligence (AI) system for cylinder rental management could be highly versatile and beneficial for various entities beyond rental companies. Some potential beneficiaries include User Industries (companies in industries like welding, food production, or manufacturing that use cylinders for industrial purposes. An advanced cylinder management system could optimize the supply chain and ensure a continuous and efficient supply), Gas Suppliers (businesses specializing in supplying industrial or medical gases could benefit from AI to optimize distribution, manage stocks, and ensure operational safety), Logistics, and Transport Services (companies in logistics and transportation could use the system to optimize delivery routes, enhance transport efficiency, and reduce operating costs), Environmental Companies (organizations involved in environmental monitoring and management could benefit from AI to ensure sustainable cylinder distribution practices and minimize environmental impact), Regulatory and Safety Entities (regulatory bodies and safety authorities could use AI to monitor and ensure compliance with safety regulations in the cylinder rental sector), Maintenance and Technical Service Companies (Businesses specializing in cylinder maintenance and repair could integrate AI for predictive and timely maintenance services), Healthcare Sector (medical facilities using gas cylinders could benefit from AI to ensure a reliable supply and effectively manage stocks, especially in critical situations), Technology and Software Companies (technology and software solution providers could develop and implement the AI-based cylinder rental system, offering an advanced and customizable platform).

Adopting an AI-based system could extend across various industries and sectors, providing tailored solutions to meet the specific needs of each entity involved in cylinder management.

6 Conclusions

This work has described and introduced an innovative system that can guarantee a breakthrough at a relatively low cost for all those industrial activities that need the monitoring and localization of a pressurized gas cylinder.

The entire system was developed using Arduino, a low-cost microcontroller, allowing the prototype phase of the control system to be further explored using a more consolidated technology in an industrial environment.

The proposed system can realize simultaneously two different important tasks: from one side, it can display the pressure level and a warning for an empty cylinder, and simultaneously, this information is automatically sent to the company that manages the charge and the substitution of the cylinders, together with the position of the empty one.

For the eventual marketing of a product, it is necessary to assess the return on investment time of the asset. In this case, evaluating the return on investment time is not easily identifiable as the variables involved are numerous and mostly tied to the management of the cylinder fleet carried out by the specific owning company. However, the system's research and development have been conducted to minimize material costs and optimize component choices to make the system commercially more appealing. The initial prototype was developed to provide an excellent solution in the cylinder rental industry. The current implementation is a preliminary step, but it is designed to allow for further improvements and adaptations based on the needs of the operating environment, including those related to artificial intelligence.

The system's modular architecture facilitates the implementation of future upgrades and the possibility of new features in response to evolving industrial needs. Of course, the focus will remain on data security and system stability to maintain industrial product standards. In conclusion, this initial prototype represents a significant step toward the realization of a complete cylinder rental system that optimizes the management and efficiency of the process for the customer's benefit while also leveraging neural networks or artificial intelligence.

The research prospects will mainly focus on refining the artificial intelligence model for parts management based on the collected data. The goal is to improve the understanding of the operating environment by improving the system's ability to dynamically adapt to changes in the demands of the cylinder model. In addition, it plans to explore opportunities to implement other emerging technologies inherent in the Internet of Things (IoT) to improve cylinder monitoring and management further. This evolution aims to create a more interconnected and responsive system, further improving a cylinder facility's overall efficiency and sustainability.

References:

[1] N. Najiid, M. Alaoui-Selsouli, A. Mahafid, An integrated production and maintenance planning model with time windows and shortage cost., *Int. Journ. Of Prod.Research,* Vol. 49, No. 8, 2011, pp. 2265-2283.

[2] A.B. Basygut, Mechanical and corrosion properties of TIG welded dissimilar AISI 420 and UNS S32205 stainless steel sheets. *Mat.Today Communications*, Vol. 33, No. 104640, 2022.

[3] N.M. Shaalan, F. Ahmed, O. Saber, S. Kumar, Gases in Food Production and Monitoring: Recent Advances in Target Chemiresistive Gas Sensors. *Chemosensors* Vol. 10, 2022, pp. 1-20.

[4] Arduino. *What is Arduino?*, [Online]. https://www.arduino.cc (Accessed Date: June 5, 2023).

[5] G. Ramesh, J. JolinDorrothi, R. Nithya Shree, Smart in sync cylinder reserving and LPG gas tracking system. *International Conference on Advances in Computing, Communication Control and Networking*, Greater Noida, India, 2021.

[6] M.S. Kumaran, P. Jayarama, R. Hounandan, B. Prahatheesh, Smart LPG cylinder monitoring and explosion management system. *International Symposium on Advanced Topics in Electrical Engineering*, Bucharest, Romania, 2021.

[7] K. P. Priya, M. Surekha, R. Preethi, T. Devika, N. Dhivya, Smart gas cylinder using embedded system *Int. J. of Inn. Res. In Electr. Electronics Instr. And Contr. Eng.*, Vol. 2, No. 2, 2014, pp. 958-962.

[8] S. Unnikrishnan, M. Razil, J. Benny, S. Varghese, C.V. Hari, LPG monitoring and

leak detection system, *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2017, pp. 1990-1993.

[9] P. Alqinsi, I. J. Matheus Edward, N. Ismail and W. Darmalaksana, IoT-Based UPS Monitoring System Using MQTT Protocols, *4th International Conference on Wireless and Telematics (ICWT)*, Nusa Dua, Bali, Indonesia, 2018, pp. 1-5.

[10] A. K. Srivastava, S. Thakur, A. Kumar, A. Raj, IoT-based LPG cylinder monitoring system, *IEEE International Symposium on Smart Electronic Systems*, Rourkela, India, 2019, pp. 268-271.

[11] A. Ahsan, M.Z. Islam, R. Siddiqua, Md.K. Rhaman, An IoT-based interactive LPG cylinder monitoring system with sensor node-based safety protocol for developing countries, *IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 2020, pp. 398-401.

[12] R. Niranjana, T. Hemadarshana, S. Ilakkya, R. Jaiveena, A. Ravi, An intelligent gas monitoring system with solenoid valve and weight cell using MQTT, *2nd International Conference on Edge Computing and Applications (ICECAA)*, Namakkal, India, 2023, pp. 1564-1570.

[13] M.W.P Maduranga, J.P.D.M. Sithara, Real-Time animal location estimation using wearable sensors and cellular mobile networks. *International Journal of Wireless and Microwave Technologies*, Vol 3, 2022, pp. 61-68.

[14] Arduino. Arduino – GPS, [Online]. https://arduinogetstarted.com/tutorials/arduino-gps?utm_content=cmp-true (Accessed Date: June 5, 2023).

[15] Arduino. Arduino GSM Shield, [Online]. https://docs.arduino.cc/retired/shields/arduino-gsm-shield (Accessed Date: June 5, 2023).

[16] Arduino. Arduino MKR Wi-Fi 1010, [Online]. https://docs.arduino.cc/hardware/mkr-wifi-1010 (Accessed Date: June 5, 2023).

[17] MQTT. MQTT: The Standard for IoT Messaging, [Online]. https://mqtt.org/ (Accessed Date: June 5, 2023).

[18] J. Fuller, GY-NEO6MV2 Flight Control GPS Module, [Online]. https://datasheethub.com/gy-neo6mv2-flight-control-gps-module/ (Accessed Date: June 5, 2023).

[19] Eric S. Raymond, NMEA Revelead, [Online]. https://gpsd.gitlab.io/gpsd/NMEA.html (Accessed Date: January 12, 2024).

[20] NMEA Data, [Online]. https://web.archive.org/web/20201020015948/http://www.gpsinformation.org/dale/nmea.htm (Accessed Date: January 12, 2024).

[21] Arduino. Securely Connecting an Arduino MKR Wi-Fi 1010 to AWS IoT Core, [Online]. https://docs.arduino.cc/tutorials/mkr-wifi-1010/securely-connecting-an-arduino-mkr-wifi-1010-to-aws-iot-core (Accessed Date: June 5, 2023).

[22] AWS. IoT Core Developer Guide, Storing device data in a DynamoDB table, [Online]. https://docs.aws.amazon.com/it_it/iot/latest/developerguide/iot-ddb-rule.html (Accessed Date: June 5, 2023).

[23] D.N. Tsyani, A. Kurniasari, C. Hudaya, Battery monitoring system with LoRA technology, *3rd International Conference on Information Technology, Information System and Electrical Engineering*, Yogyakarta, Indonesia, 2018, pp. 125-129. 19

[24] Loganathan Nachimuthu, Reem Ahmed Al-Mahrouqi, Tamadhar Salim Al-Abri, Muna Al-Hadrami, *IOT based Energy Monitoring for Practical Loads using NodeMCU*, WSEAS Transactions on Electronics, vol. 14, pp. 49-56, 2023, doi: 10.37394/232017.2023.14.6

[25] AWS. AWS IoT Core, [Online]. https://aws.amazon.com/it/iot-core/ (Accessed Date: June 5, 2023).

[26] Grafana Lab, [Online]. https://grafana.com/products/cloud/?src=ggl-s&mdm=cpc&camp=b-grafana-exac-emea&cnt=118483912276&trm=grafana&device=c&gclid=Cj0KCQjw06-oBhC6ARIsAGuzdw16VvHye-XWBwkmYwQcGuz07myUDwnmisiyFvSyiYbINOtrA0L1O94aAiUGEALw_wcB (Accessed Date: June 5, 2023).

[27] Hassan Ali Al-Ababneh, Victoria Borisova, Alina Zakharzhevska, Polina Tkachenko, Natalia Andrusiak, Performance of Artificial Intelligence Technologies in Banking Institutions, *WSEAS Transactions on Business and Economics*, vol. 20, pp. 307-317, 2023, https://doi.org/10.37394/23207.2023.20.29.

[28] Reem Al-Araj, Hossam Haddad, Maha Shehadeh, Elina Hasan, Mohammad Yousef Nawaiseh, The Effect of Artificial Intelligence on Service Quality and Customer Satisfaction in Jordanian Banking Sector, *WSEAS Transactions on Business and Economics*, vol. 19, pp. 1929-1947, 2022, https://doi.org/10.37394/23207.2022.19.173.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.