International Journal of Science and Engineering Applications
Volume 15-Issue 04, 75 – 78, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1504.1013

Design and Implementation of an IoT-Based Real-Time Wellbore Liquid Level Monitoring System Using Python

Zixin Chen
School of Electronic Information and Electrical Engineering
Yangtze University
Jingzhou, China

Abstract: To enhance the real-time monitoring capabilities of oilfield submergence, ensure operational safety, and optimize oil well production efficiency, this paper designs and implements a dynamic fluid level measurement and control software using the Python programming language and the acoustic wave method. The developed software enables real-time monitoring and processing of critical information from multiple oil wells, including dynamic fluid level depth, echo propagation time, and operating voltage. Furthermore, the system supports data access via mobile terminals, allowing for ubiquitous monitoring of well conditions and timely detection of operational anomalies. Experimental results demonstrate that the software accurately acquires downhole fluid level data, with the absolute error controlled within 1.0 meter and the average relative error as low as 0.1%. The Python-based measurement and control software significantly improves the real-time performance, accuracy, and convenience of monitoring, providing robust technical support for the safe and efficient development of oilfields.

Keywords: oil well dynamic fluid level; acoustic wave method; real-time monitoring; Python; measurement and control software; data acquisition; Internet of Things (IoT)

1. INTRODUCTION

Accurate monitoring of the "dynamic fluid level" is essential for ensuring operational safety and production efficiency during oil well extraction. However, complex downhole environments and data transmission latencies often lead to inaccurate interface determination, posing risks like equipment damage. While advancements in automated control have improved measurement consistency, selecting an optimal detection methodology remains critical.

Current detection methods include pressure sensing, the float method, and the acoustic wave method. The acoustic wave method is preferred for its high precision, non-intrusive nature, and adaptability across diverse well conditions. To process the resulting logging data, various software platforms have been employed. LabVIEW-based systems, though capable in signal processing, suffer from limited portability and complex version control due to their graphical programming nature. Similarly, C#-based systems often face longer development cycles and a smaller ecosystem of specialized libraries.

In contrast, Python offers an extensive library ecosystem and superior cross-platform compatibility, making it an ideal platform for modern intelligent logging. Consequently, this paper develops a Python-based measurement and control software that leverages the acoustic wave method. This system integrates high-precision detection with the flexibility of the Python ecosystem, significantly enhancing software development efficiency and portability.

2. MEASUREMENT PRINCIPLE AND METHODOLOGY

The propagation of acoustic waves within a wellbore is illustrated in Figure 1. An acoustic wave generator at the wellhead emits pulses that travel downward along the casing-tubing annulus. Reflections occur when the waves encounter tubing collars, generating "collar echoes" that are captured by the surface receiver. The remaining wave energy continues to propagate until it reflects off the gas-liquid interface, producing a distinct "liquid level echo". Based on this principle, the dynamic fluid level depth h is determined by the product of the acoustic propagation velocity v and the travel time t, as expressed in Equation (1)

$$h = \frac{vt}{2} \tag{1}$$

where h is the dynamic fluid level depth, v is the acoustic velocity in the wellbore gas, and t is the total propagation time. By analyzing these propagation characteristics, this methodology provides an effective and accurate means of identifying the fluid interface position.

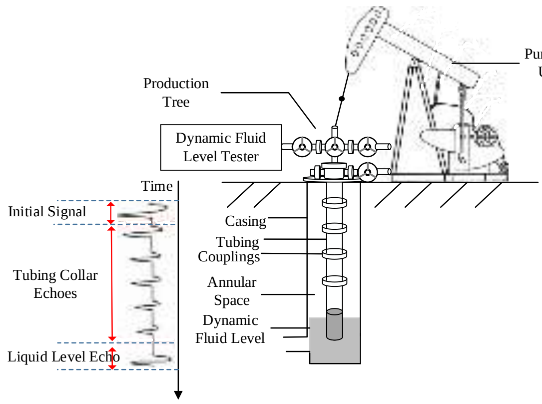

Figure. 1 Schematic Diagram of Acoustic Wave Propagation

3. SOFTWARE SYSTEM DESIGN

3.1 Development Environment

Python was selected as the primary programming language due to its extensive library ecosystem, cross-platform compatibility, and high efficiency in scientific computing. The software was developed within the PyCharm integrated development environment (IDE), leveraging its advanced code analysis and debugging tools to ensure robust system performance.

3.2 Overall System Architecture

To achieve efficient data acquisition, processing, and management, this paper proposes a multi-layered software framework. The overall architecture is illustrated in Figure 2.

The system workflow initiates with the transmission of raw acoustic data from logging instruments to the Alibaba Cloud platform via the MQTT protocol. Subsequently, the Data Acquisition Module retrieves the dynamic fluid level data and device parameters, forwarding them to the central server. In the Processing Module, Python-based algorithms are employed to perform signal analysis and extract key features from the raw data, ensuring high data integrity and accuracy. Finalized datasets are then transmitted to the Storage Module, where they are maintained in a highly structured database for efficient retrieval.

This architectural design ensures that the software provides users with real-time, reliable monitoring and control services by enabling flexible data management and rapid response to downhole anomalies.

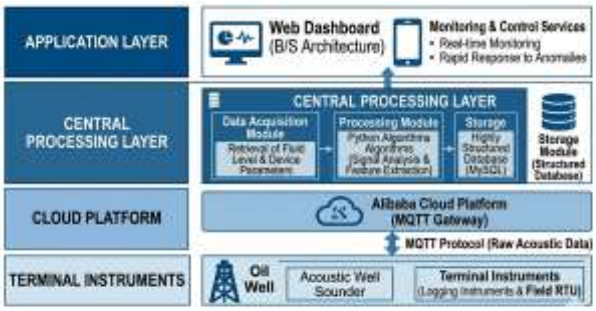

Figure 2: Overall System Architecture

3.3 Functional Components of the Software

The measurement and control software is categorized into four primary functional modules: data acquisition, data processing, data storage, and data management, as illustrated in Figure 3. The Data Acquisition Module is responsible for harvesting raw fluid level signals and device status parameters. The Data Processing Module parses complex JSON payloads and extracts characteristic features. Subsequently, the Data Storage Module executes database schema mapping for persistent storage. Finally, the Data Management Module facilitates real-time visualization and trend analysis.

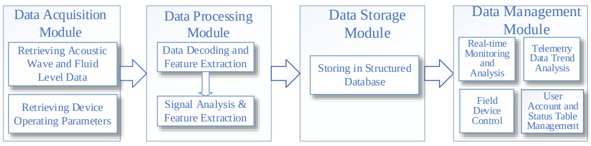

Figure 3: Functional Component Framework

3.4 Data Acquisition Module

The acquisition module performs synchronized collection of wellbore acoustic data and electrical parameters. It supports localized detection and real-time display, which are essential for field installation and diagnostic calibration. Additionally, a configurable scheduling engine allows users to define detection intervals based on reservoir production rates.

The data transmission architecture follows the uplink workflow shown in Figure 4. Field instruments communicate with onboard 4G modules via universal asynchronous receiver-transmitter (UART) interfaces. By establishing a persistent connection with the cloud server and publishing to specific MQTT topics, the captured data is routed through a message consumer group for server-side processing, ensuring a reliable foundation for subsequent analysis.

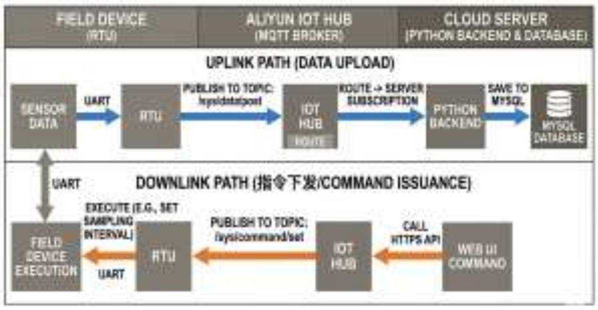

Figure 4: IoT Bidirectional Data Flow

3.5 Data Processing Module

This stage involves the algorithmic parsing of unstructured JSON data. The system extracts critical metadata, including Device ID, operating voltage, and wellhead pressure. These parameters are essential for contextualizing the fluid level measurements.

The processing core generates dynamic waveform visualizations to represent the acoustic reflection process, including the raw signal, tubing collar echoes, and the fluid interface reflection. By analyzing these trends, the software provides a comprehensive profile of the wellbore environment. Concurrently, processed data is formatted into a digital dashboard featuring key metrics: dynamic fluid level depth, annulus pressure, ambient temperature, echo travel time, and acoustic velocity.

3.6 Data Storage Module

A MySQL-based relational database is utilized for structured data persistence. The schema is optimized using three specialized tables: User Account Management, Telemetry Storage, and Device Configuration. This modular storage approach enhances query performance and system flexibility. Once processed, data is mapped to the corresponding fields in the Telemetry table according to strict data types. This structured back-end provides a robust service layer for the high-level management application.

3.7 Data Management Module

The management interface adopts a B/S (Browser/Server) architecture, implemented via a custom-built Web application. This interface allows for ubiquitous access to real-time data, device status, and user privileges. The B/S structure eliminates the need for localized client software, enabling remote monitoring through standard web browsers.

Furthermore, the system facilitates bidirectional control through the downlink path shown in Figure 4. The server issues control commands via HTTPS API calls to the IoT platform, which then publishes the data to specific device-end topics. This framework ensures that remote parameter configurations are executed with high reliability and low latency.

International Journal of Science and Engineering Applications
Volume 15-Issue 04, 75 – 78, 2026, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1504.1013

4. APPLICATION AND RESULTS

4.1 Web Interface and Functional Verification

The proposed Python-based software integrates data acquisition, signal processing, and database management into a unified Web-based dashboard. Figure 5 illustrates the user login interface, which ensures secure data access through credential authentication. The administrative dashboard, shown in Figure 6, provides comprehensive data visualization and authority management. Administrators can configure user permissions (highlighted by the yellow dashed box in Figure 6) to maintain system security. Simultaneously, critical telemetry such as echo travel time, casing pressure, and acoustic velocity are displayed in real-time (indicated by the red dashed box), allowing for immediate diagnostic analysis.

Figure 5: User login interface

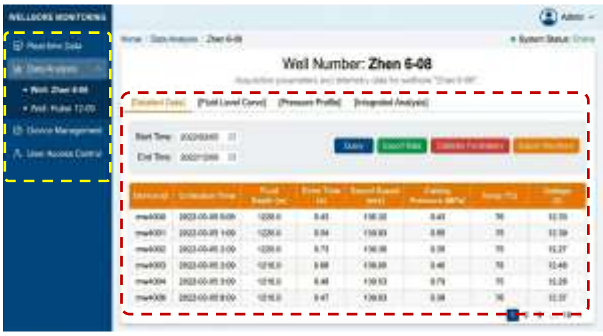

Figure 6: Administrative dashboard

4.2 Field Testing and Accuracy Analysis

To validate the reliability of the software in real-world scenarios, field tests were conducted at an active oil well. Table 1 presents a subset of telemetry data captured on August 24, 2025, with a sampling interval of 30 minutes.

Table 1. Representative Data Captured During Software Testing

Test Count	Collection Time	Fluid Level Depth (m)	Echo Time (s)	Sound Speed (m/s)	Casing Pressure (MPa)
1	14:52:08	1057.40	6.80	310	0.32
2	15:22:57	1059.27	6.83	310	0.34
3	15:52:03	1063.45	6.86	310	0.36
4	16:22:03	1070.43	6.91	310	0.41
5	16:52:03	1074.15	6.93	310	0.44
6	17:22:03	1072.91	6.92	310	0.48
7	17:52:02	1074.31	6.93	310	0.50
8	18:22:02	1077.10	6.95	310	0.53
9	18:52:04	1082.06	6.98	310	0.55
…	…	…	…	…	…

Table 2. Comparison Between Manual Measurements and Software Results

Test Count	Collection Time	Measured Depth (m)	Software Depth (m)	Abs. Error	Rel. Error
1	2025/09/06 14:28	1198.36	1199.41	1.05	0.09%
2	2025/09/08 10:14	1225.84	1224.94	-0.90	0.07%
3	2025/09/13 15:13	1238.58	1237.62	-0.96	0.08%
4	2025/09/17 11:13	1249.84	1250.77	0.93	0.07%
5	2025/09/20 10:13	1259.69	1258.63	-1.06	0.08%
6	2025/09/22 15:42	1268.98	1269.84	0.86	0.07%

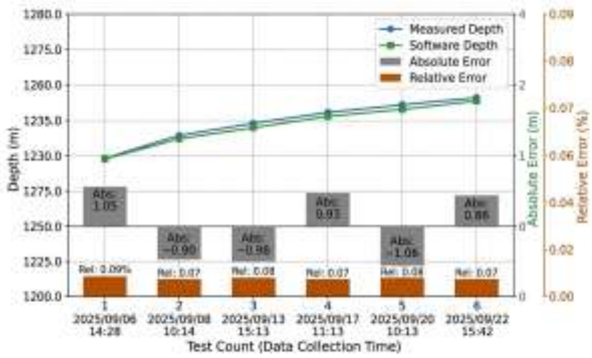

Figure 7: Accuracy Comparison: Manual vs. Software

The software's measurement accuracy was evaluated by comparing its outputs with manual field measurements. As shown in Table 2, the absolute error is consistently maintained within pm1.0 m, with an average relative error of only 0.1%.

These results, further visualized in Figure 7 (Depth Variance Comparison), demonstrate that the software provides high-precision data acquisition and stable performance under field conditions. The implementation of this system effectively reduces data transmission latency and enhances the predictive maintenance capabilities of oilfield operations.

5. CONCLUSION

This paper developed a dynamic fluid level measurement and control software leveraging the Python programming language and the acoustic wave method. The system achieves a seamless integration of real-time monitoring, cloud-based data storage, and remote Web-based visualization. Field applications verify that the software significantly improves the efficiency of oil extraction operations and reduces maintenance costs by providing accurate and timely wellbore data.

While the current system demonstrates high reliability and precision (average relative error of 0.1%), future research will focus on integrating advanced machine learning algorithms to enhance the system's capability for anomaly detection and predictive modeling in complex downhole environments.

REFERENCES

[1] Jia, W., Zhou, W., and Li, T. F. 2014. A Review of Dynamic Fluid Level Detection for Oil Well. *Applied Mechanics & Materials*, 456, 582-586.

[2] Qi, L. Q., Huang, Q. L., and Zhang, Y. F. 2022. Design of Multi-terminal Monitoring System for Oil Well Dynamic Fluid Level Based on Internet of Things and Cloud Platform. *Information Recording Materials*, 23(06), 24-27.

[3] Wei, Y., Cheng, J., Wang, L. P., et al. 2023. Research Progress on Two Key Issues in Dynamic Fluid Level Depth Detection. *Science Technology and Engineering*, 23(20), 8473-8483.

[4] Wang, Y. X., Wei, Y., Wang, L. P., et al. 2022. Design and Implementation of Downhole Dynamic Fluid Level Depth Measurement System. *Instrument Technique and Sensor*, (10), 81-87.

[5] Cui, X. M., Wei, Y., Guo, T., et al. 2022. Design of Acoustic Signal Simulator in Dynamic Fluid Level Detector. *Foreign Electronic Measurement Technology*, 41(01), 57-62.

[6] Chen, D. F., Han, X. L., and Yang, J. 2008. Discussion on Detection Methods of Oil Well Fluid Level. *Oil & Gas Well Testing*, (02), 60-61+78.

[7] Fang, J. 2016. Design and Application of Real-time Signal Processing Software Based on LabVIEW for Measurement While Drilling Surface System. *Petroleum Tubular Goods & Instruments*, 2(03), 24-27.

[8] Yin, D. W., Cai, Y. X., Shu, S. L., et al. 2022. Control System of Swept-source OCT Imaging Based on LabVIEW. *Experimental Technology and Management*, 39(05), 118-122.

[9] Guan, X., Wang, K. S., and Song, C. L. 2020. Design and Implementation of Oil Well Dynamic Fluid Level Monitoring System. *Journal of Xi'an Shiyou University (Natural Science Edition)*, 35(03), 122-126.

[10] Huangfu, W. H. 2018. Research on the Application of Oil Well Dynamic Fluid Level Detection System Based on Internet of Things. *Master's Thesis*, *Xi'an Shiyou University*.

[11] Jiang, Q. L. and Zhang, F. 2023. Research on Method of Rapidly Obtaining Web Page Data Based on Python and Requests. *Modern Information Technology*, 7(16), 100-103.

[12] Hou, L. 2020. Progress and Trends of International Logging Technology in 2020. *World Petroleum Industry*, 27(06), 49-54.

[13] Wang, Q. Y. and Huang, Q. Z. 2021. Analysis of Research Progress of Machine Learning in the Field of Petroleum Logging. *Cleaning World*, 37(03), 120-122+124.

[14] Cheng, X., Zhou, J., Fu, H. C., et al. 2023. Applicability and Application of Machine Learning Algorithms in Geophysical Logging. *Northwestern Geology*, 56(04), 336-348.