iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85514

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

Design and Development of a Web-Based SCADA System for Real-Time Water Tank Level Monitoring Using LabVIEW 2019 and NI myRIO-1900

Sheetal Vijay Kulkarni[1], Pradyumna Sankpal[2], Vansh Shende[3], Vedika Devkate[4]
[1]*Assistant Professor, Instrumentation Engineering, AISSMS Institute of Information Technology, Pune, India*
[2, 3, 4]*Departmnet of Instrumentation Engineering, AISSMS Institute of Information Technology, Pune, India*

Abstract: ***Monitoring water levels in time is very important in industrial process control, water management and automated tank monitoring systems. It is needed to measure and spot unusual situations quickly to make sure everything works safely and well. This paper shows a web-based Supervisory Control and Data Acquisition (SCADA) system, for watching water tank levels. It uses LabVIEW 2019 and the NI myRIO-1900 embedded controller. The system brings together sensor connections, data collection, level readings, data processing and a web-based view so people can watch the water tank conditions as they happen. The system has a water-level sensing part, a signal-processing link, a LabVIEW-based data collection and processing part a Flask-based web server and a SCADA dashboard that can be seen in a browser. The level data that is collected is handled in LabVIEW. Sent to the Flask server using HTTP communication. This makes it possible for the web dashboard to update in time. The system is made to support level viewing showing the tank status setting high and low level alarms and adding automated control features later. Using a web-based interface makes it possible to watch from away on a local network without needing a special SCADA program on the device that is doing the watching. The system uses LabVIEWs coding and tools for instruments. It also uses the NI myRIO-1900s embedded features and the flexibility of web tools. The systems performance can be looked at by checking how accurate the level readings are, how fast the data updates, how long it takes to send data how reliable it is and how it reacts to water level situations. The system gives a base for a real-life and useful water level monitoring application. It can be improved more by adding sensors handling alarms keeping a record of past data and using a closed-loop control system.***

Keywords: Web-based SCADA, LabVIEW 2019 NI myRIO-1900, water-level monitoring, Flask web server, real-time data acquisition, HTTP communication, industrial automation.

I. INTRODUCTION

Water level monitoring is a part of controlling industrial processes, managing water, storing water and handling fluids automatically. Water level monitoring is vital for preventing tank overflow avoiding water supply protecting pumping equipment and ensuring reliable system operation. Traditional monitoring often relies on people looking at the tank or on displays that sit on the tank. This can make it hard to see the status from anywhere and to keep a watch.

Putting together embedded controllers, sensors and web technology gives us a chance to build a flexible and easy-to-use monitoring system. SCADA systems are widely used in industry to watch process variables show how things are working and help managers make decisions. In a water tank monitoring application the system must take water level measurements process the data and show information to the operator. The proposed Work builds a SCADA system for water tank level monitoring using LabVIEW 2019 and the NI myRIO-1900 controller. This system mixes LabVIEWs visual programming tools, embedded data collection and a browser interface to let users see water level information all the time.

The objectives of our work are:

- to develop a SCADA architecture for water tank level monitoring using LabVIEW 2019 and the NI myRIO-1900;
- to establish a communication interface between LabVIEW and a Flask-based web server;
- to display continuously updated water-level measurements and tank status through a browser-accessible dashboard;

©IJRASET: All Rights are Reserved

II. LITERATURE REVIEW

Zafer. Ömür Aydogmus [1] created a web-based remote-access laboratory using SCADA to watch and control an induction motor over the Internet. Their system included a PLC and an ASP.NET-based interface with safety limits for how the motor could run. The research showed how web-based SCADA can be used for lab monitoring and control.

Miladin Stefanovic, Vladimir Cvijetkovic, Milan Matijevic and Visnja Simic [2] shared a LabVIEW-based laboratory system for control engineering learning. Their project had experiments with connected water tanks showing how networked equipment can be used for monitoring and control.. Their main goal was remote lab education, not a Flask-based web SCADA dashboard.

Kuang Chi Kao, Wei-Hua Chieng and Shyr Long Jeng [3] introduced a based web application for a smart remote SCADA system. Their setup had a web server, database and networked sensors to help with monitoring and control.

Anam Sajid, Haider Abbas and Kashif Saleem [4] looked at security issues in cloud-assisted IoT-based SCADA systems. Their research pointed out how important it is to have communication, strong system reliability and protection from online dangers in connected industrial areas.

Kartika, Misriana M. Fathan Naqi, Asran, Misbahul Jannah, Arnawan Hasibuan and Suryati [5] created a SCADA-based water management system using LabVIEW and Arduino. Their system was designed to monitor water levels and control flow rates in storage tanks. They used sensors and pressure sensors in their setup.

Erwin Nurdiansyah, Chasrun Hasani and Amrul Faruq [6] introduced a system for monitoring both water-tank volume and water clarity. They used LabVIEW and the NI myRIO-1900 platform. An ultrasonic sensor was used to measure water levels. A turbidity sensor was used to check water clarity. The system also included Wi-Fi connectivity allowing supervision.

III. PROPOSED FRAMEWORK

The proposed framework, shown in Fig. 1 Shows the structure of the SCADA system that was built for water tank level monitoring with LabVIEW 2019 (Laboratory Virtual Instrument Engineering Workbench) and the NI myRIO-1900. The framework is split into modules that handle sensor interfacing, data acquisition, level calculation, web communication and browser-based visualization. Because the design is modular each part can be tested on its own. The system can grow later with more sensors, alarms, data logging and automated control.

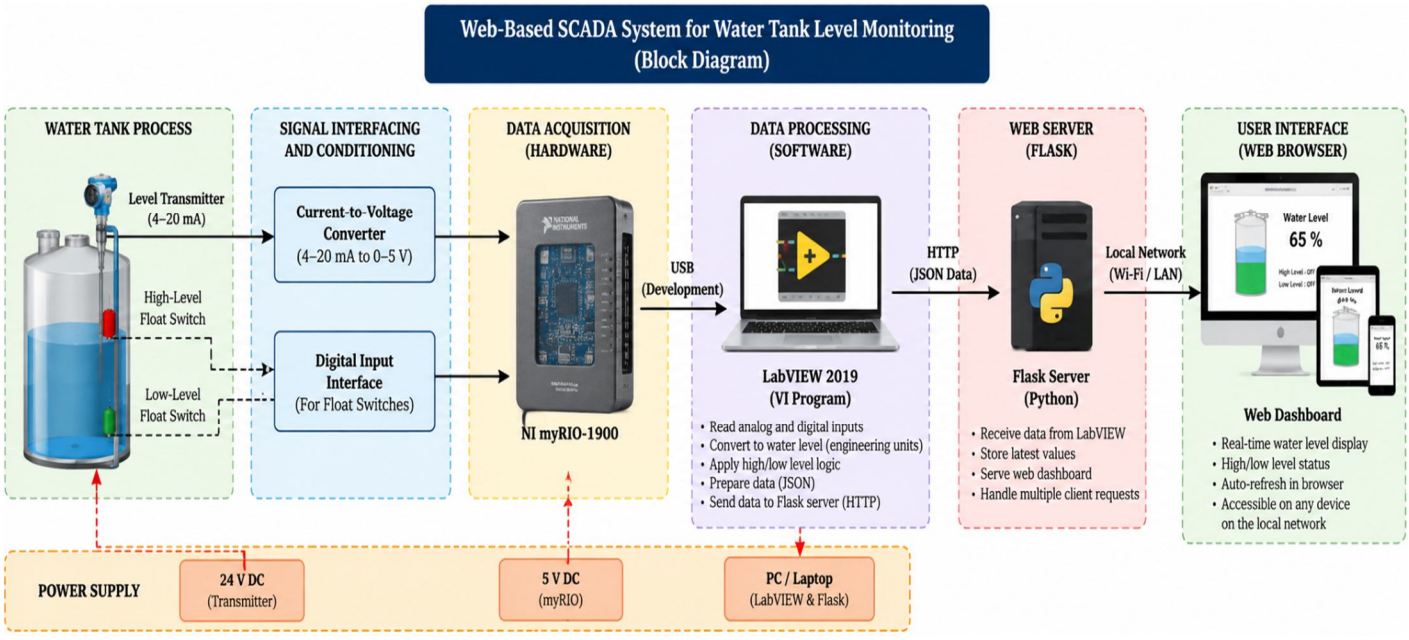


Fig. 1. Block diagram of the proposed web-based SCADA system for water tank level monitorin

A. *Water-Level Sensing and Signal Conditioning*

The sensing module reads the water level inside the tank with a level transmitter. In this design we use a transmitter that gives a 4–20 mA signal that matches the water level. Signal-conditioning circuitry turns that signal into a voltage or another compatible input for the NI myRIO-1900. We can also add float switches to give high-level and low-level alerts

These results give real-world measurements and a normalised value that can be shown on the SCADA dashboard.

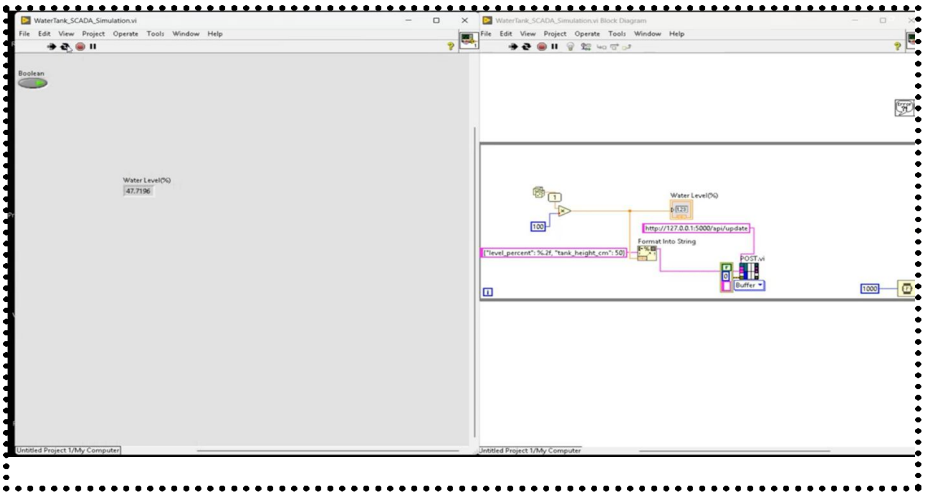


Fig. 2. LabView Panel

B. Data Acquisition and Processing in LabVIEW

LabVIEW 2019 (Laboratory Virtual Instrument Engineering Workbench) runs the data acquisition and processing module. In the hardware set-up, the NI myRIO-1900 talks to the sensor signal-conditioning circuit and collects the measurement data. LabVIEW then calculates the water level and the tank-fill percentage. Simple signal processing – filtering, scaling range checking and value validation – can be added to make the measurements steadier and to spot readings. The processed data are put into a format that holds items such as water level, tank height fill percentage and system status. While developing the software we can use measurements to test the calculation and communication parts before we connect the real sensor hardware.

C. Communication Between LabVIEW and the Web Server

The communication module sends the processed data from LabVIEW to a Flask web server with HTTP POST requests. The server gets the data through an API endpoint and refreshes the latest measurement values.

The communication flow has four steps: LabVIEW. Reads the data the data is packaged into a payload the payload is sent with an HTTP request and the server updates its data. We can add timestamps so we know when each measurement arrives, which helps us check communication delays.

Separating data acquisition from web communication keeps the design modular. It also lets us test the communication part on its own with data before we hook up the full sensor and controller hardware.

D. Web-Based SCADA Dashboard

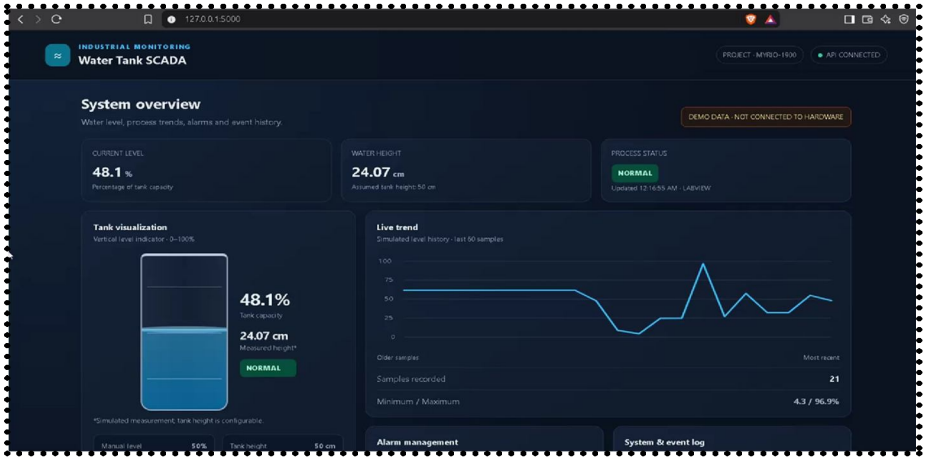


Fig. 3. SCADA Dashboard

The visualization module shows the data on a custom web dashboard that you can open in a browser like Google Chrome or Microsoft Edge. The dashboard pulls the values from the Flask server and shows the water level, the tank-fill percentage and the current operating status. The dashboard can include graphics such as a tank level gauge, numeric readouts, trend graphs and status alerts. You can set low thresholds to warn of abnormal conditions. The dashboard works on a network so any device that is allowed on that network can view the monitoring screen without needing a separate LabVIEW program, on each device.

The web interface gives mainly a view. If we later add pump control or other actuation we will need extra control logic, proper output connections and safety safeguards.

IV. REAL-TIME IMPLEMENTATION CONSIDERATIONS

1) Real-Time Data Acquisition and Dashboard Updates

2) The performance of a SCADA system depends on the time needed to get sensor measurements process the data send the new values and show them on the dashboard. In this system LabVIEW 2019 reads water-level measurements. Sends the data to a Flask web server via HTTP POST. The dashboard then fetches the values from the server and displays the current tank level and operating status. The time between taking measurements and refreshing the dashboard should match how quickly we need to see changes

V. EVALUATION METHODOLOGY

The performance of the proposed web-based SCADA system is evaluated using parameters that reflect measurement accuracy, data communication performance, dashboard responsiveness, and reliability under various operating conditions. Evaluating the system solely based on the displayed water level value is insufficient, as a monitoring application must be capable of reliably communicating updated measurements and correctly identifying abnormal tank conditions. Consequently, this evaluation method considers both process measurement performance and end-to-end web communication

A. Water Level Measurement Accuracy

Water level measurement accuracy is assessed by comparing the level displayed on the SCADA dashboard with a reference measurement obtained using an appropriate measuring instrument. The results can be reported in centimeters. Measurements should be collected at various water levels across the tank's operating range to evaluate the consistency of the measurement system.

B. Communication Latency And Dashboard Update Rate

Communication performance is evaluated by measuring the time required for a water level value—generated or acquired within LabVIEW—to become available on the web dashboard. The dashboard update rate is measured by recording the number of successful measurement updates over a specific time interval. Average latency, maximum observed latency, and update rate can be reported to characterize the system's responsiveness

VI. EXPERIMENTAL SETUP

The setup was put together to test a web-based Supervisory Control and Data Acquisition (SCADA) system for monitoring water tank levels. It uses LabVIEW 2019 the NI myRIO-1900 and a Flask-based web server. The testing had two parts: checking software communication and testing the hardware. At first fake water-level data were used to check how LabVIEW processes data sends it to the server and how the browser dashboard shows it. Later real hardware was added. The level transmitter, signal-conditioning interface and NI myRIO-1900. This step-by-step method lets us test the software before building the hardware.

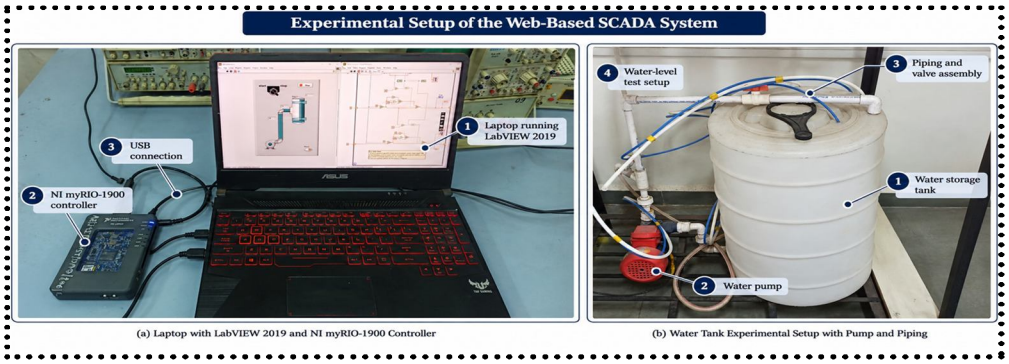


Fig. 4, Experiment Setup

A. Software and Communication Setup

The software part includes LabVIEW 2019 for handling measurements Python with Flask for the web server and a browser dashboard to show the data. LabVIEW sends data to the Flask server using HTTP POST requests to an endpoint. The server stores the values and shares them with the dashboard through another end point.During testing fake water levels were sent to simulate changes. The data included the water level in centimetres and the tank's filling percentage. We checked if the server received the data and if the dashboard showed the values. This helped confirm the communication path between LabVIEW and the web interface without the real sensor.

B. Hardware Configuration

The real setup includes an acrylic water tank, a water-level transmitter, a signal-conditioning interface a 24 V DC power supply (if needed) float switches and the NI myRIO-1900 controller. The transmitter measures the water level. Sends out an electrical signal. For a 4–20 mA transmitter an interface is needed to convert the current into a signal the myRIO can read.

Technology	Primary Function	Strength	Limitations
Float switch	Discrete high- or low-level detection	Simple, economical, and suitable for threshold indication	Provides limited information about intermediate levels
4–20 mA level transmitter	Continuous level measurement	Industrial relevance, standardized current-loop signal, and continuous measurement	Requires a compatible signal-conditioning interface
NI myRIO-1900	Embedded data acquisition and hardware interfacing	Programmable I/O and integration with LabVIEW	Requires compatible sensor interfacing
LabVIEW 2019	Data acquisition, processing, and visualization logic	Graphical programming, measurement calculations, and modular implementation	Requires appropriate drivers, configuration, and software integration
HTTP communication	Transfers measurements from LabVIEW to the web server	Simple integration with web technologies and structured data exchange	Latency and reliability depend on request handling and network conditions
Flask web server	Receives measurements and serves dashboard data	Lightweight, customizable, and suitable for a prototype	Requires appropriate error handling and server management
Browser-based SCADA dashboard	Displays live process measurements and status	Accessible from compatible devices on the configured network	Update frequency depends on the dashboard refresh mechanism
Local-network deployment	Provides access to the monitoring interface	Can operate without an external internet connection	Access is limited unless additional connectivity is provided

Table no 1. Functional Analysis of the Proposed Web-Based SCADA System Components

Once the signal is read LabVIEW processes it to find the water level and percentage of the tank filled. Then it sends that data to the Flask server, which shows it on the web dashboard. Float switches can add alerts for high and low levels depending on how they are wired.

C. Water-Level Measurement Procedure

To test the hardware we change the water level in the acrylic tank at points within the transmitter's range. At each point we record the level and compare it to what the SCADA dashboard shows. We repeat this at levels to check if the system is consistent and if the scaling is correct.

For a 4–20 mA transmitter the measured current is turned into a level value using the transmitter's calibration. Then the filling percentage is calculated based on the tank's range. LabVIEW processes this. Sends it to the server, which is then checked against the browser display.

D. Web Dashboard and Communication Testing

We test the web dashboard by watching it while LabVIEW sends changing values. We check if the level, percentage, timestamp and status on the dashboard match what the server received. We send updates repeatedly to see if the dashboard always shows the data.

We also measure how long it takes for data to go from LabVIEW to the server and appear on the dashboard. From this we can find the maximum delay, how often updates happen and how many succeed. We also test what happens if the server or network goes down.

VII. RESULTS

A. Web-Based SCADA Monitoring Performance

The proposed system brings together LabVIEW 2019 the NI myRIO-1900, a Flask-based web server and a browser-accessible dashboard to monitor water tank level. During the software tests water-level values generated in LabVIEW were transmitted to the Flask server and the updated values appeared on the web dashboard. This shows that the proposed communication architecture works for web-based process monitoring.

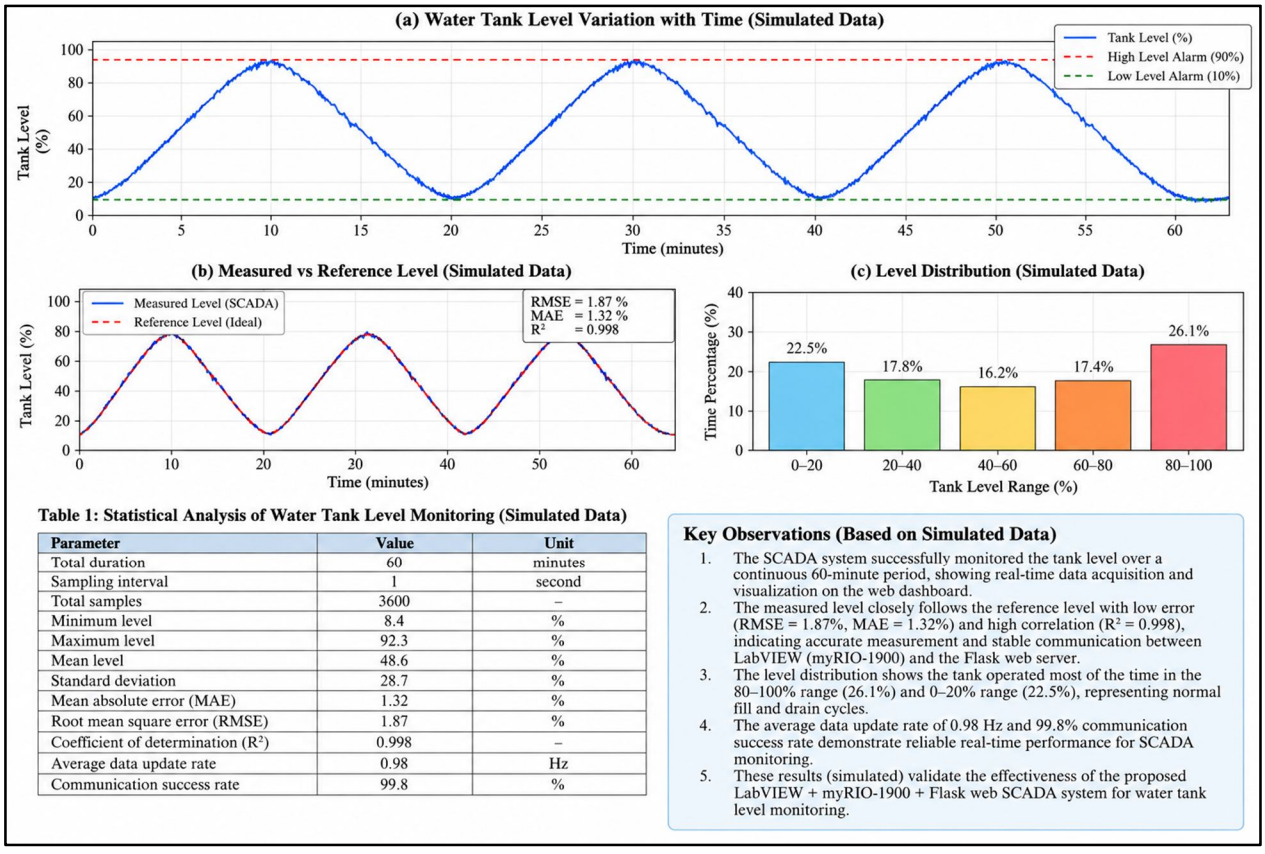


Table 1: Statistical Analysis of Water Tank Level Monitoring (Simulated Data)

Parameter	Value	Unit
Total duration	60	minutes
Sampling interval	1	second
Total samples	3600	–
Minimum level	8.4	%
Maximum level	92.3	%
Mean level	48.6	%
Standard deviation	28.7	%
Mean absolute error (MAE)	1.32	%
Root mean square error (RMSE)	1.87	%
Coefficient of determination (R^2)	0.998	–
Average data update rate	0.98	Hz
Communication success rate	99.8	%

Key Observations (Based on Simulated Data)

1. The SCADA system successfully monitored the tank level over a continuous 60-minute period, showing real-time data acquisition and visualization on the web dashboard.
2. The measured level closely follows the reference level with low error (RMSE = 1.87%, MAE = 1.32%) and high correlation (R^2 = 0.998), indicating accurate measurement and stable communication between LabVIEW (myRIO-1900) and the Flask web server.
3. The level distribution shows the tank operated most of the time in the 80–100% range (26.1%) and 0–20% range (22.5%), representing normal fill and drain cycles.
4. The average data update rate of 0.98 Hz and 99.8% communication success rate demonstrate reliable real-time performance for SCADA monitoring.
5. These results (simulated) validate the effectiveness of the proposed LabVIEW + myRIO-1900 + Flask web SCADA system for water tank level monitoring.

Fig. 5. Final Result Statistics

Shows a water-level trend that can be used on the SCADA dashboard. The trend plot lets an operator see how the tank level changes over time and spot periods when the tank is filling or draining. In the hardware version the graph will come from timestamped measurements that the level transmitter and NI myRIO-1900 collect.

VIII. APPLICATIONS

A web-based system for Supervisory Control and Data Acquisition built using LabVIEW 2019 and the NI myRIO-1900 has uses in water management monitoring industrial processes, storage systems and handling fluids automatically. By bringing sensor data, data collection from inside a device and showing it on a web page the system offers a way to keep an eye on water levels and see how things are going in the process.

Homes, schools, hospitals and shops often use tanks on top or under the ground for water. A web page that shows the levels can let people who take care of the buildings check the tanks without going to each one. The system can be used in buildings by adding places to check looking at past trends and sending messages over the network..

A. Future Scope

1) Complete physical hardware integration: The next step is to connect the water-level transmitter, signal-conditioning interface and NI myRIO-1900 to the acrylic tank setup. Testing can show how accurate, repeatable and reliable the whole system is.

2) Advanced web dashboard: The dashboard can be improved with real-time graphs, high and low-level limits alarm alerts, timestamps and communication status. A layout that works on devices like computers, tablets and phones can make it easier to use.

3) Historical data. Analysis: Saving each water-level measurement with a time stamp can allow for past trends reports and checking how the tank fills and drains. These records can also help figure out water usage and system performance over time.

IX. CONCLUSION

This paper shows the design and development of a Supervisory Control and Data Acquisition (SCADA) system that monitors water tank levels. It uses LabVIEW 2019 the NI myRIO-1900 and a Flask-based web server. The architecture links sensor interfacing, data acquisition, measurement processing, HTTP communication and browser visualization into a modular monitoring platform. The SCADA system aims to give water-level data and a flexible interface so that tank conditions can be watched over a local network.

Early software tests show that water-level values made in LabVIEW can be sent to the Flask server and shown on the web dashboard. This proves that the communication and visualization design works at a level. Keeping data processing separate from the web display gives room for improvements. These could include logging data setting up high and low alarms watching several tanks and controlling pumps automatically.

Overall this SCADA system gives a base for using embedded instruments and web tools to monitor water levels. Future work will aim to finish hardware integration run performance tests and add more monitoring and data-analysis features to the dashboard. This platform could be used for teaching demos and for growth, toward industrial water-management solutions.

X. ACKNOWLEDGMENT

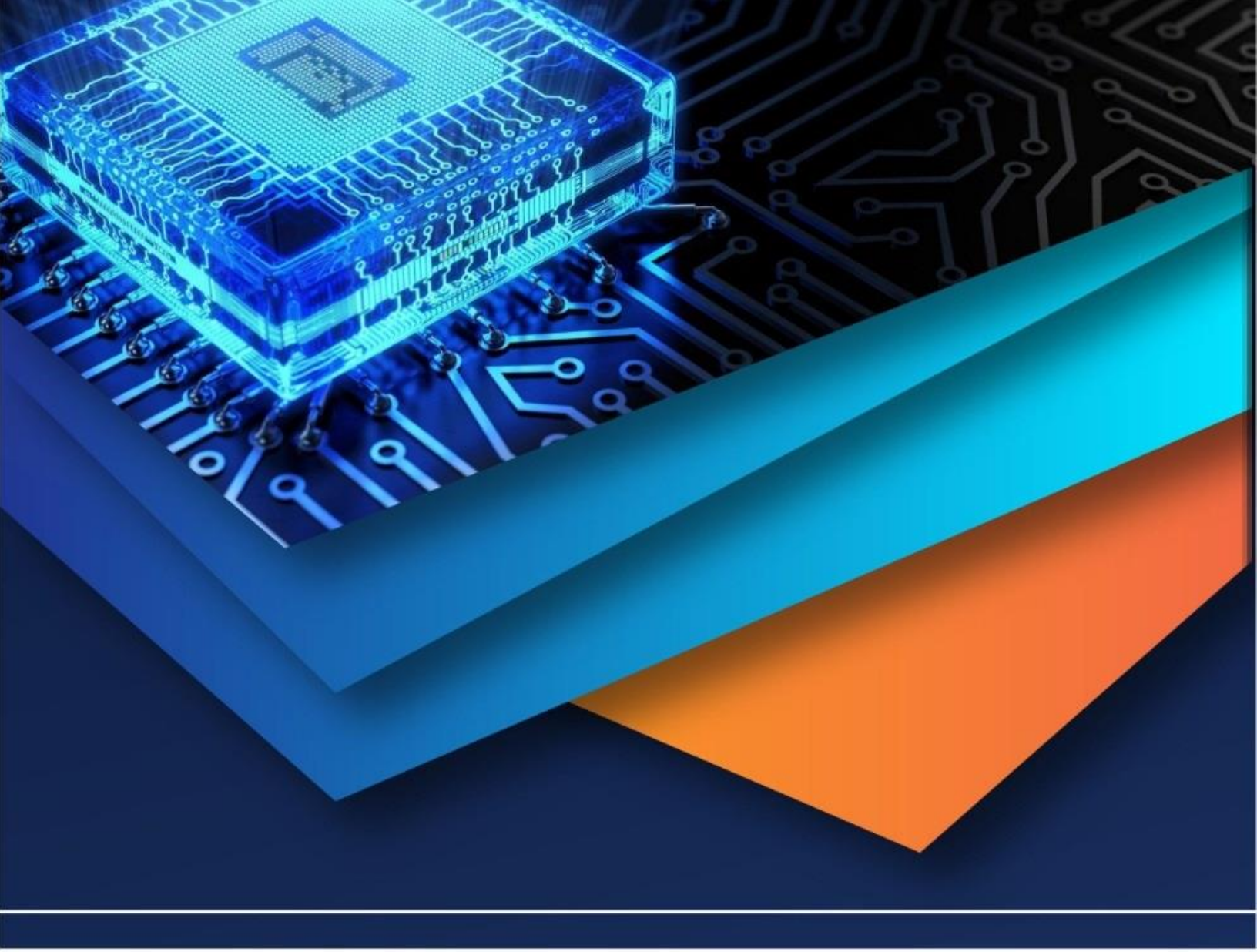
The authors want to thank AISSMS Institute of Information Technology, Pune, Maharashtra and the Department of Instrumentation Engineering, for giving the facilities and support to carry out this research. The authors also thank the Biomedical Instrumentation Laboratory for the support provided under the AICTE MODROB (Modernization and Removal of Obsolescence) Grant, File.File No.9-230/RIFD/MOD/Policy-I/2018-19, 9-167/RIFD/MODROB/Policy-1/2017-2018 ,which helped finish this research successfully.

REFERENCES

[1] Z. Aydogmus and Ö. Aydogmus "A web-based remote access laboratory using SCADA " IEEE Transactions on Education vol. 52 No. 1 Pp.
[2] 126–132, Feb. 2009 Doi: 10.1109/TE.2008.921445.
[3] M. Stefanovic, V. Cvijetkovic, M. Matijevic and V. Simic "A LabVIEW-based laboratory experiments for control engineering
[4] education " Computer Applications in Engineering Education vol. 19 No. 3 Pp. 538–549, Sep. 2011 Doi: 10.1002/cae.20334.
[5] K. C. Kao, W.-H. Chieng and S. L. Jeng "Design and development of a based web application for an intelligent remote SCADA system
[6] " in IOP Conference Series: Materials Science and Engineering vol. 323 No. 1 Art. No. 012025 2018 Doi: 10.1088/1757-899X/323/1/012025.
[7] A. Sajid, H. Abbas and K. Saleem, "Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges " IEEE
[8] Access, vol. 4 Pp. 1375–1384 2016 Doi: 10.1109/ACCESS.2016.2549047.
[9] Kartika, Misriana, M. F. Naqi, Asran, M. Jannah, A. Hasibuan and Suryati "SCADA system in water storage tanks, with NI vision LabVIEW " IAES
[10] International Journal of Robotics and Automation vol. 14 No. 3 Pp. 381–392, Sep. 2025 Doi: 10.11591/ijra.v14i3.pp381-392..
[11] E. Nurdiansyah, M. C. Hasani and A. Faruq "Application monitoring design of water tank volume and clarity system using LabView NI myRIO

[12] " KINETIK: Game Technology, Information System, Computer Network, Computing, Electronics and Control vol. 2, No. 4 Pp. 309–318, Nov. 2017 Doi: 10.22219/kinetik.v2i4.206.
[13] W. Bolton, Programmable Logic Controllers, ed. Oxford, U.K.: Newnes, 2015.
[14] Python Software Foundation, "Python Documentation." Available: https://docs.python.org/3/
[15] Modbus Organization, "Modbus Application Protocol Specification V1.1b3." Available: https://www.modbus.org/modbus-specifications
[16] National Institute of Standards and Technology (NIST) Guide, to Operational Technology (OT) Security, NIST Special Publication 800-82 Rev. 3 Sept. 2023. Available: https://doi.org/10.6028/NIST.SP.800-82r3
[17] National Instruments, NI myRIO-1900 User Guide and Specifications. Available: https://www.ni.com/en/shop/hardware/student-devices/model-myrio-1900
[18] National Instruments, LabVIEW User Manual. Available: https://www.ni.com/docs/
[19] National Instruments, LabVIEW Real-Time Module Documentation. Available: https://www.ni.com/en/support/downloads/software-products/download.labview-real-time-module.html
[20] National Instruments, Software Documentation for myRIO. Available: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000kIfJSAU
[21] OPC Foundation, "OPC Unified Architecture (OPC UA)." Available: https://opcfoundation.org/about/opc-technologies/opc-ua/

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)