Open Source IoT-Based SCADA System for Remote Oil Facilities Using Node-RED and Arduino Microcontrollers

Charles Aimiuwu Osaretin
Ph.D. Candidate
ECE, Memorial University
St. John's, NL, Canada
caosaretin@mun.ca

Mohammad Zamanlou
M.Eng. Student
ECE, Memorial University
St. John's, NL, Canada
mzamanlou@mun.ca

M. Tariq Iqbal
Professor
ECE, Memorial University
St. John's, NL, Canada
tariq@mun.ca

Stephen Butt
Professor
PE, Memorial University
St. John's, NL, Canada
sdbutt@mun.ca

***Abstract*—An open source and low-cost Supervisory Control and Data Acquisition System based on Node-RED and Arduino microcontrollers is presented in this paper. The system is designed for monitoring, supervision, and remotely controlling motors and sensors deployed for oil and gas facilities. The Internet of Things (IoT) based SCADA system consists of a host computer on which a server is deployed using the Node-RED programming tool and two terminal units connected to it: Arduino Uno and Arduino Mega. The Arduino Uno collects and communicates the data acquired from the temperature, flowrate, and water level sensors to the Node-Red on the computer through the serial port. It also uses a local liquid crystal display (LCD) to display the temperature. Node-RED on the computer retrieves the data from the voltage, current, rotary, accelerometer, and distance sensors through the Arduino Mega. Also, a web-based graphical user interface (GUI) is created using Node-RED and hosted on the local server for parsing the collected data. Finally, an HTTP basic access authentication is implemented using Nginx to control the clients' access from the Internet to the local server and to enhance its security and reliability.**

Keywords*—SCADA, Internet of things, Node-RED, Nginx, IoT-Based SCADA.

I. Introduction

SCADA stands for Supervisory Control and Data Acquisition. It is a general term used to refer to groups of hardware and associated software devices that enable the supervision and control of operational processes, typically in manufacturing plants and industrial settings. The collection and aggregation of data are followed by processing, logging, and visualization of the collected data. The instrumentation and control of critical operational parameters in most oil and gas production processes are made possible by deploying field instrumentation devices such as transmitters, receivers, sensors, actuators, pumps, and valves. The collection of data and communication of corrective action to field instrumentation devices is made possible by the deployment of microprocessors/microcontrollers in the SCADA system. Programmable logic controllers (PLC) and terminal units (TUs) help to route the collected data through communication channels to the central SCADA computers called master terminal units (MTUs). The software on MTUs provides a suitable human-machine interface (HMI) for analysis, interpretation, and visualization of process variables. Indicators and alerts also help operators to identify and react to trigger events [1-4]. Traditional SCADA deployed in oil field operations are generally limited in terms of upgrading, programmability, and efficiency of network administration. Regular SCADA systems are also relatively expensive, preventing their adoption in low flowrate wells. The SCADA architecture employed in this work is a fourth-generation SCADA based on the Internet of Things framework. By deploying software and electronics to sensors embedded in critical processes, distributed computing resources in the IoT-based SCADA can aggregate, log, process, and visualize the data collected either on-site or remotely using internet connectivity. The core of the IoT-based SCADA server proposed in this paper is the Node-RED server deployed on a local computer [5-7].

II. Literature Review

Production from oil wells driven by artificial lift is complicated and requires instrumentation, metering, supervision, and control. The level of fluid produced requires close monitoring to optimize fluid level in the tank between the upper and lower limit and hence minimize or prevent leakages, spills, and environmental pollution [8]. Flowrate monitoring also helps to estimate the gross volume of produced fluids and to compare with flowrates recorded downhole for early detection of wax formation. Measurement and control of temperature at different nodes in the production process directly impacts the phase behavior and flow assurance of the produced fluids. Critical interventions, such as chemical injection, dewaxing, heating, and production adjustment, require temperature monitoring for flow assurance and asset integrity [9]. Industrial automation and IoT technology provide a sustainable, secure, and scalable alternative for low-cost monitoring, supervision, and control of several operational parameters in remote oil and gas operations [10-12]. Data acquisition, data logging, aggregation, and visualization are generally required for making important decisions, hence the adoption of SCADA systems.

In terms of licensing, there are two classes of SCADA, proprietary and open source system. In open source, the system is entirely built from scratch using modularized and often freely licensed applications. It is beneficial because adding and replacement of components is cheaper and more manageable without the need for special expertise and training. By using open standards, interoperability is enhanced, and the system lifecycle cost can be reduced. On the contrary, a proprietary or "closed" system is developed and managed by a vendor, with upgrade and modifications dependent on the availability of the specific vendor and the license stipulating the terms and conditions of service [13].

Although cloud server architecture is increasingly deployed for hosting, logging, aggregation, storage, supervision, and control of data received by embedded sensors, cybersecurity is a major concern, as leveraging of cloud architecture exposes the system to cyberattacks with a significant risk of data loss, data corruption or disruption of service [12]. This work adopts a local on-premises server running Node-RED for collection, aggregation, logging, monitoring, and control of data. Remote access to the local server is controlled by setting up an authentication request for username and password. The internet protocol (IP) address of the server is also concealed from the client by using a reverse proxy server called Nginx. It enhances the security and reliability of the local server; hence by running the application on an on-premise machine, the user can take advantage of most of the required features of cloud-based architecture without long term subscription costs and reduced risk of cyberattacks [14-16].

III. Experimental Design of Proposed IoT-Based Scada System

The case study is a low flowrate oil well driven by an electric motor and powered by solar energy. Figure 1 shows the schematic of the project in the case study. The scenario consists of a single well

978-1-7281-8416-6/20/$31.00 ©2020 IEEE

artificially lifted with a jack pump. In this scenario, the produced fluid is collected and accumulated into a single tank. The essential nature of oil production requires continuous site monitoring to acquire, log, and process the data so that timely intervention can ensure continuous and optimal operation.

Accurate tracking of the flowrate from the tubing and monitoring of the tank level is also needed for logging daily, monthly, and annual production from the well of interest. Logging and appropriately tracking the fluid flowrate could lead to early detection of sand and wax in the tubing. The temperature of the produced fluid at the wellhead and the storage tank will also need to be closely monitored to avoid wax formation and deposition. Maintenance planning and early fault detection are also considerations that justify this research for remote low flowrate oil wells.

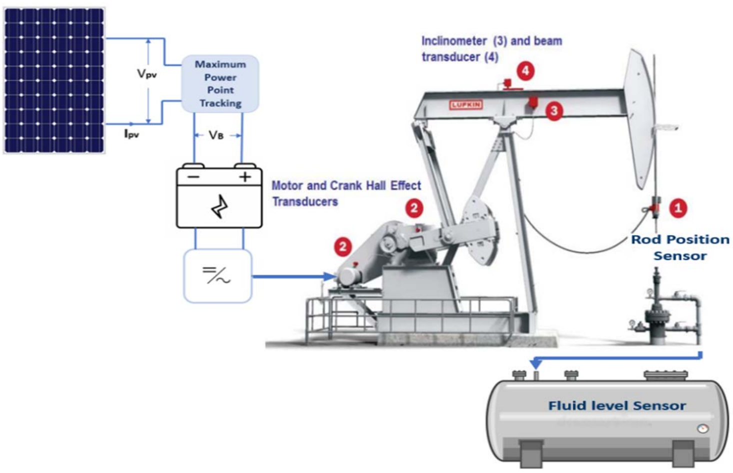

Fig. 1. Schematic diagram of a case study showing elements of the proposed system.

The IoT-based open source SCADA system consists of three subsystems, a master terminal unit and two terminal units, as shown in figure 2.

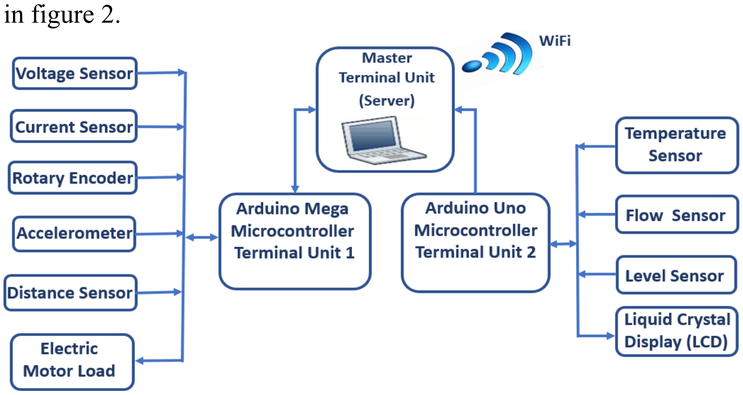

Fig. 2. Schematic diagram of the proposed IoT-based SCADA System

The design and implementation of the SCADA system entail hardware and software selection. At the subsystem level, the sensor hardware is connected to the Arduino microcontrollers through base shields and breadboards; each sensor is then programmed in the Arduino integrated development environment (IDE). The decision to adopt two terminal units stems from the need for redundancy and the limitation of available input and output Arduino nodes in Node-RED.

In terms of read/write capability of the Arduino pins, Node-RED can manage digital and analog input and outputs, PWM, and servo over the Arduino pins via the use of both Firmata protocol and serial communication. The design methodology is to develop a web application that serves as both a dashboard and a graphical user interface (GUI), enabling an operator to remotely read, log, and monitor field sensor conditions over the Internet.

IV. Implementation Methodology

The proposed SCADA system consists of a master terminal unit (MTU) and two other terminal units (TUs). Each instrumentation device is connected to the terminal units (Arduino microcontrollers) through base shields and breadboards. The Arduino microcontrollers collects the data and messages; acting as clients to the master terminal unit, which aggregates and processes the received data as a host. The USB connection is adopted as the communication channel between MTU and TUs. Serial communication is used in the Arduino Uno, while the Arduino Mega microcontroller implements the Firmata protocol in its firmware. Standard Firmata is used on the Arduino Mega for the connected accelerometer, distance, position, voltage, and current sensors. Therefore, using the Firmata protocol, the Node-RED application on the host server is able to interact with the Arduino Mega and hence directly access the respective pins of the Arduino Mega, altering the state, mode, and reading the desired signals from the connected sensors. The design methodology is to develop a web-based application that serves as a human-machine interface (HMI) for visualization and enables an operator to remotely read, log, and monitor field sensor conditions over the Internet.

A. *Master Terminal Unit (MTU)*

The MTU is a local server that hosts the Node-RED application. It also logs and aggregates the sensor data. In addition to the dashboard, a graphic user interface is also hosted on the MTU for real-time visualization. The MTU controls and influences the operation of the terminal units, which are directly connected to the field instrumented devices (sensors, actuators, and on-site display). It logs, aggregates, and publishes sensor data from the IoT devices onto a web interface for visualization and interpretation.

B. *Terminal Units (TUs)*

The Arduino microcontrollers used in this work are both connected to the host server on the computer via a USB port. The Arduino microcontrollers model or represent the field-deployed terminal units. Arduino Mega is used as the first terminal unit (TU1). In this subsystem, the voltage, current, rotary, accelerometer, and distance sensors are all connected to the Arduino mega and read on the local server through Node-RED using Firmata. Also, the motors are controlled through Node-RED using Arduino nodes and Firmata. Arduino Uno is used as the second terminal unit (TU2). In this subsystem, an Arduino sketch is uploaded to read the fluid level, flowrate, and temperature sensors' values connected to the Arduino Uno, display them on the on-site LCD, and send them to the Node-RED local server via the serial port connection.

C. *Voltage Sensor Module*

The sensor uses the principle of a voltage divider to measure an input source voltage. Due to its simple design, the voltage sensor is low-cost, an output voltage (Vout) from 0 to 5V is produced and supported by the Arduino. In this work, the signal pin (S) of the voltage sensor is connected to the analog pin A9; the plus and minus sign pins are connected to the 5V and GND pins, respectively on the Arduino. GND and the input pins of the sensor are connected in parallel across the output of the solar panel to measure the PV output voltage.

D. *ACS 723 Hall-Effect current sensor*

ACS 723 is a high precision current sensor that is economical and fully integrated. The principle of "Hall-Effect" is employed. Such that when a single supply voltage VCC of (4.5 to 5.5V) is applied across the inputs: IP+ and IP- (copper conduction path), a resulting current flow is produced, which generates a magnetic field. The magnetic field is sensed by the Hall-Effect IC to produce a proportional current.

Hence there is electrical isolation between the sensed circuit and the sensing circuit. A high-power sensed circuit arrangement with either DC or AC sources is thus compatibly sensed by a low power sensing circuit. The two input pins: VCC and GND of this sensor are connected in series to the solar panel, while the output signal is read from the analog pin A10 on the Arduino Mega.

E. *Rotary Position Sensor*

The rotary sensor turns continuously and hence behaves like a potentiometer. It is also called the shaft encoder, enabling the precise measurement of motors' rotation, with less temperature drift and higher precision. The rotary sensor used in this work produces an analog signal value based on the angular displacement or motion of the shaft with an output voltage range from 0 to 5V DC. It is used as a position monitoring device in this work. The VCC and the GND pins of this sensor go to the supply and ground of the Arduino Mega respectively, and the output signal is read by the analog pin A8 of the Mega.

F. *ADXL335 Accelerometer*

The accelerometer works on the principle of the piezoelectric effect. The ADXL335 is a low cost, low power, and highly integrated 3-axis accelerometer that measures both static and dynamic acceleration. Static acceleration is the acceleration due to gravity, which enables the device to be used as a tilt sensor, hence enabling the device to detect free fall and measure tilt. In contrast, dynamic acceleration results from motion, vibration, or shock. In this work, the output signals are analog voltages from three pins X, Y, and Z, which are proportional to the acceleration and directly integrated with the analog pins A13, A12, and A11, respectively.

G. *SharpIR GP2Y0A21YK0F Distance Sensor*

The Sharp infrared sensor is a low-cost distance measuring sensor unit of the analog output type. It is an integrated unit that combines an infrared emitting diode with a position-sensitive detector and signal processing unit. The detectable distance for this sensor ranges from 10 - 80cm. This device can also be used as a proximity sensor and outputs a voltage corresponding to the detection distance. The VCC and GND pin of this sensor is connected to the supply and ground of the Arduino Mega, respectively, while the Vout signal is read by the analog pin A7 of the Mega.

G. *DC Motor*

The prime movers on the Arduino Mega (controller 1) are two DC gear motors. They are connected to the Arduino Mega via the Arduino motor shield, which models the variable frequency drive as implemented. The Arduino motor shield REV3 is based on the L298P motor controller chip and contains a dual full-bridge driver. The base shield has an operating voltage range of 0.5-12V and is powered by a 9V regulated power supply from the DC solar panel. The shield provides a suitable hardware interface between the Arduino mega and the DC motors for independent direction control, speed control, and braking. The pins D3, D9, and D12 of the Arduino Mega are used for driving Motor A, and the pins D8, D11, and D13 of the Arduino Mega are used for driving Motor B.

H. *Fluid Level Sensor*

The Grove fluid level sensor is a low power sensor that adopts the principle of capacitance and 2, 8-bit microcontroller units: ATTINY1616 MCUs. It consists of capacitive pads embedded on a printed circuit board and sealed to be completely waterproof. It measures the water level up to 10 cm. The analog input from the capacitors is converted into a digital signal by three (3) comparators. It communicates over the I2C bus of the Arduino and is typically deployed in water level sensing applications. VCC and the GND pins of this sensor go to the supply and ground of the Arduino Uno, respectively, and I2C serial communication is used.

I. *Flowrate Sensor*

The YF-S401 is a low-cost fluid flow sensor with a flow range of 0.3-6 L/min. It consists of a Hall effect sensor and a fluid rotor enclosed in a plastic valve casing. Fluid flowing through the rotor causes a magnetic rotor to spin. The resulting rotating magnetic field cuts across a solenoid wire, causing a current flow and producing a voltage at the terminals of the Hall effect sensor. The resulting voltage is proportional to the rate of flow of fluid across the rotor and hence serves as a suitable measure of flowrate. The output pulse width signal of the sensor is connected to the microcontroller. The signal output pin is connected to the digital pin D3 and is programmed to trigger the interrupt 1 on the Arduino Uno to calculate the flowrate.

J. *Temperature Sensor*

The Grove temperature sensor V1.2 is used in this research. It has an operating voltage range from 3.3 to 5V and detects from -40 to 125ºC. The ambient temperature is detected by the chip. The resistance of a thermistor will decrease when the ambient temperature increases. This relationship is useful for predicting the ambient temperature with an accuracy of ± 1.5ºC and is suitable for the application. Vout signal is read by the analog pin A0 of the Arduino Uno.

K. *Liquid Crystal Display (LCD)*

The LCD is used as a local display to show the temperature value. It is connected to the Arduino Uno via the I2C bus. The LCD is useful for reading and displaying temperature value on-site. Also, the flowrate and the water level can be displayed here.

L. *Microcontroller 1 (Arduino mega)*

The first terminal unit (TU1) is essentially Arduino Mega with Firmata through serial port connection to the Node-RED installed on the local server computer. The voltage, current, rotary encoder, accelerometer, and distance sensors are read by the microcontroller making five (5) field instrumented devices connected to TU1, as shown in figure 2. Besides, the two motors, A and B, are connected to the Arduino Mega via a motor shield and are controlled through Node-RED.

M. *Microcontroller 2 (Arduino Uno)*

The second terminal unit, TU2, is an Arduino Uno with serial port connection to the Node-RED running on the local server. Four field instrumented devices are connected to TU1, including three sensors (temperature, flow, level) and a liquid crystal display (LCD). The advantage of using the serial connection in TU2 is that it permits the creation and execution of functions in the Arduino IDE, allowing for complex computations to be executed within the code and only the desired outputs printed over the serial ports.

N. *Node-RED IoT Platform on the Local Server*

The flow of data between interconnected devices is implemented in Node-RED, a web-based programming tool. By wiring nodes together, JavaScript objects and functions are created, flows are built and instantly deployed in Node-RED which consists of a node.js-based runtime. The graphic user interface (GUI) is also available in Node-RED and contains a dashboard with logs and charts for visualization of sensor data in real-time [14,15]. The design methodology is to provision the web server on the local computer and make it accessible from the Internet, hence client requests to the Node-RED application running on the webserver are transferred from the external IP address, and port of the router to the internal IP address and port of the local server (port forwarding) and server responses are in turn transferred from the webserver to the router (network address translation, NAT) and on to the respective client. Hence port forwarding is implemented by mapping the internal IP address/port of the server to that of the router. This option exposes the Node-RED server on the Internet for remote access. After that, a basic access authentication is implemented using Nginx. Nginx requires the internet client to provide a username and password before access to the Node-RED on the server is granted hence improving the security, as shown in figure 3 [17,18]. The process flows for position, current, voltage, accelerometer, fluid level, flowrate, temperature, as well as electric motor controls and distance sensors are shown in figures—4 to 9, respectively.

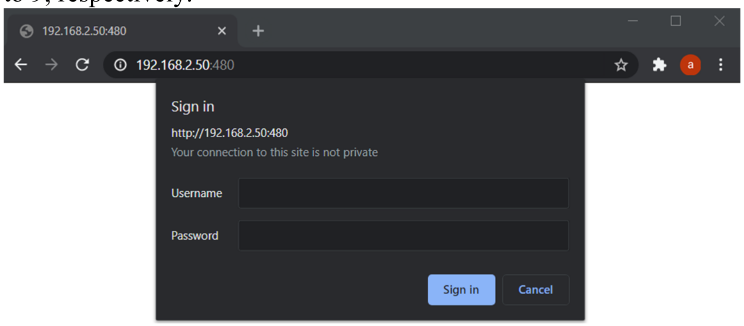

Fig.3. NGINX client access authentication page when logging in to the server.

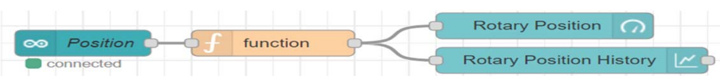

Fig.4. Process flow for the Position sensor.

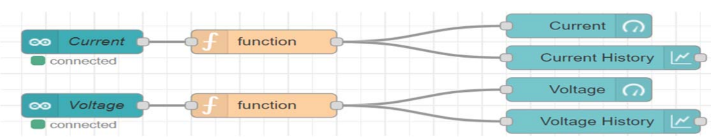

Fig.5. Process flow for the current and voltage sensor.

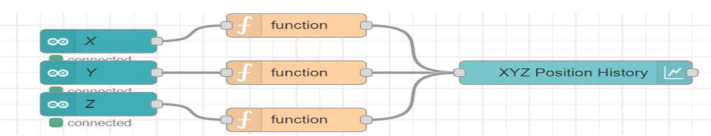

Fig.6. Process flow for the accelerometer.

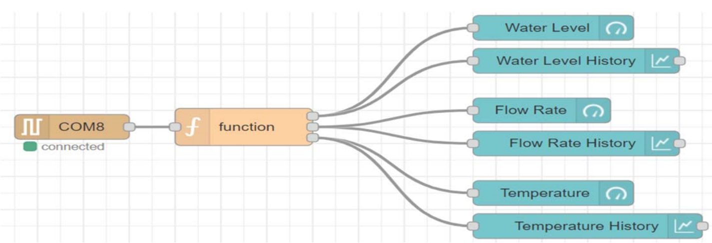

Fig.7. Process flow for the fluid level, flowrate and temperature sensor.

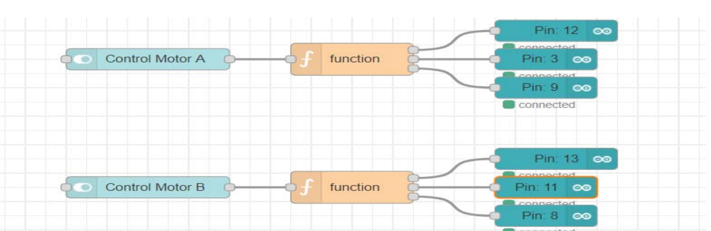

Fig. 8. Process flow to control motors A and B.

Fig.9. Process flow for the distance sensor.

V. Experimental Setup /Results

The experimental setup of the IoT-based SCADA hardware circuit is shown in figure 10. The Arduino Mega and Uno are connected to the computer via USB ports. In this experimental setup, all the sensors, motors, and display are laid out according to figure 2.

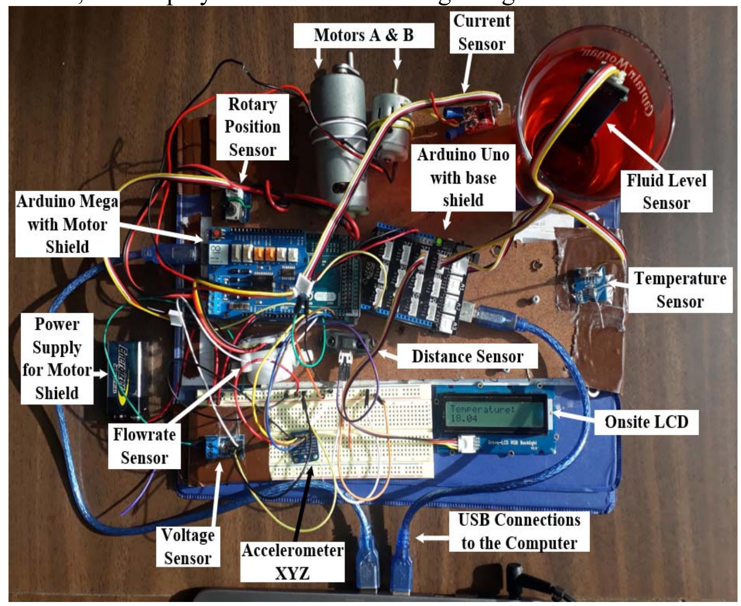

Fig.10. Experimental setup of the proposed IoT-based SCADA system

The results are shown by subsystems as follows:

Subsystem 1: The dashboard for sensors connected to Arduino Mega is shown in figures 11 and 12. The dashboard for this subsystem includes gauge and chart outputs for current, voltage, distance, accelerometer and rotary position sensors. In this work, the user interface shows the received sensor data, both as a gauge and a chart.

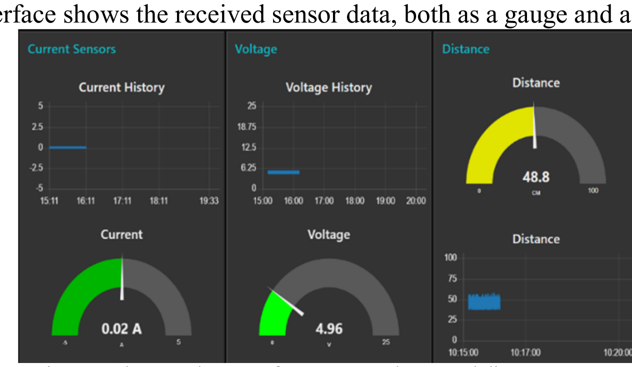

Fig.11. Charts and gauges for current, voltage, and distance sensors.

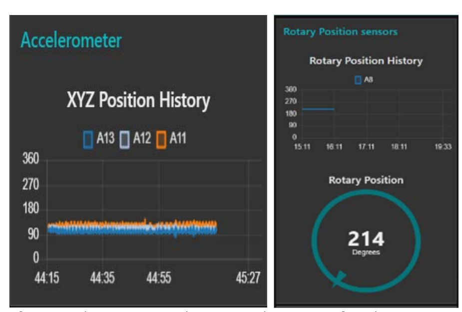

Fig.12. Charts for accelerometer, charts, and gauges for the rotary position sensor.

Subsystem 2: The dashboard for the sensors on Arduino Uno is shown in figure 13. It is available in Node-RED as a web GUI that updates or responds in real-time. The user interface shows the sensor data both as a gauge and a chart. It shows the water level, flow rate, and temperature readings.

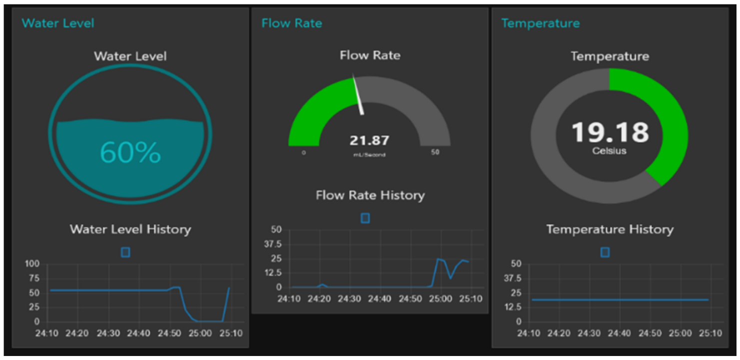

Fig.13. Charts and Gauges for water level, flowrate, and temperature sensor.

C. *Subsystem 3*: Figure 14 shows the dashboard panel to control motors A and B connected to Arduino Mega's motor shield. This dashboard uses Arduino nodes in Node-RED and Firmata on the microcontroller to start and stop the motors.

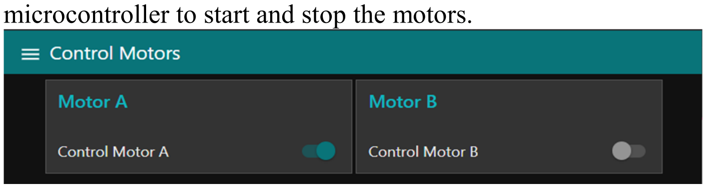

Fig.14. Dashboard panel to control motors A and B

VI. Discussion

Monitoring of the oil well site parameters such as flowrate from the producing well and comparing it with the speed of the electrical motor is imperative to diagnose problems in an actual producing well. When the speed of the electric motor is high, and the flow rate of the produced fluid is declining, based on the thresholds set, it could be an indicator of fluid pound, gas lock, rod failure or wear and tear in the subsurface rod pump. It could also point to sand production at the producing interval or issues in reservoir production. Early-onset of flow assurance problems such as wax and paraffin formation could also be detected by tracking and monitoring the behavior of the voltage and current drawn by the electric motor and mapping trends, which could be further corroborated with flow rate and cross-referenced with downhole temperature and pressure gauges. When the IoT devices are mounted on an actual 3D model and the data acquired is logged and integrated appropriately, these sensor trends could be aggregated into historical data and analyzed for prediction of failure or monitoring production performance.

VII. Conclusion

A low-cost IoT-based SCADA system is presented in this work; the Arduino Mega and Uno microcontrollers are deployed as terminal units to communicate with and aggregate data from a total of eight (8) sensors, namely the water level, flowrate, temperature, current, voltage, distance, rotary position, and accelerometer sensors. The electric motor and the LCD on-site display are also transducers whose operation and outputs are controlled by the MTU through the terminal units. Designing and securely architecting networks that interconnect these field instrumented devices in the industrial Internet of Things framework is required for remote sensing, instrumentation, and control of critical field parameters.

VIII. Future Work

This research provisioned a server instance on a local computer which hosts both the Node-RED application and graphic user interface on its web browser. In the future, the authors intend to build the Node-RED application as a cloud-native application in IBM Watson. The role of the rotary encoder will also be expanded beyond simple monitoring to automated speed control of the electric motor for direct regulation of motor speed and indirect control of fluid production rate. Email alerts and notification will also be adopted. In terms of security; port forwarding and basic authentication were used in this research, secure sockets layer (SSL) will be further adopted in to encrypt HTTP connection to HTTPS, ensuring that responses to client requests are redirected to a secure version of the Node-RED application running on the provisioned server. The intent is to ultimately mount the sensors on a 3D model of a sucker-rod pump and use the sensor readings in combination with artificial intelligence and machine learning to calibrate a digital twin.

References

[1] L. D. Cuayo, J. Kerbee Culla, J. Gualvez, S. E. Padua and R. John Gallano, "Development of a Wireless Microcontroller-based SCADA RTU, "*TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea (South), pp. 2566-2570, 2018.

[2] L. O. Aghenta and M. T. Iqbal, "Low-Cost, Open Source IoT-Based SCADA System Design Using Thinger.IO and ESP32 Thing". Electronics. 8. 822. 10.3390/electronics8080822, 2019.

[3] M. Zamanlou and M. T. Iqbal, "Development of an Economical SCADA System for Solar Water Pumping in Iran," IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS), 2020.

[4] N. Lal and A. Purohit, "Data Acquisition of Solar Power Plant Using Scada System," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 23, no. 4, pp. 189-194, 2015.

[5] http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-123.html

[6] https://insights.sei.cmu.edu/sei_blog/2018/03/12-risks-threats-vulnerabilities-in-moving-to-the-cloud.html

[7] https://www.sierrawireless.com/applications/energy/oil-and-gas/

[8] https://www.scadalink.com/products/satscada/single-well-oil-battery-monitoring-system/

[9] C. B. Cardoso, I. N. Alves, G. S. Ribeiro,"Management of Flow Assurance Constraints". Offshore Technology Conference, 2003.

[10] A. Adeyinka, A. Tsakporhore, A. Oduwole, S. Jumbo, F. Odusote, S. Aliyev, T. Eme, "Re-positioning Mature Fields for Sustainability: Deriving Additional Value by Applying the Internet of Things via Remote Well Monitoring," Society of Petroleum Engineers, 2020.

[11] W. G. Elmer, & J. B. Elmer. "Pump-Stroke Optimization: Case Study of Twenty-Well Pilot," Society of Petroleum Engineers. 2018.

[12] A. Vijay, and V. S. Unni, "Protection of Petroleum Industry from Hackers by Monitoring and Controlling SCADA System," Society of Petroleum Engineers, 2012.

[13] https://electrical-engineering-portal.com/three-generations-of-scada-system-architectures\

[14] "Node-RED," https://nodered.org/

[15] M. Lekic, G. Garadasevic, "IoT sensor integration to NodeRED platform," East Sarajevo, Infoteh-Jahorina, 2018.

[16] S. Chanthakit and C. Rattanapoka, "MQTT Based Air Quality Monitoring System using Node MCU and Node-RED," 2018 Seventh ICT International Student Project Conference (ICT-ISPC), Nakhonpathom, pp. 1-5, 2018.

[17] www.sitepoint.com/configuring-Nginx-ssl-node-js/

[18] https://www.Nginx.com/resources/glossary/Nginx/