International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 09 | Sep 2023 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of Multi-SCADA System for Drone-Based Reliability Response

Liliana Loughlin

Department of Robotics Engineering,
Worcester Polytechnic Institute, Worcester, MA USA

---***---

Abstract *A machine-controlled reliability response system comprised of sensors, mobile platforms, communications with data storage and analysis components has been developed and deployed. The implementation of a Supervisory Control and Data Acquisition (SCADA) system control is necessary in order to effectively transport bi-directional data between the project's core elements such as sensor and actuator equipped drones. A description of how this is achieved through the combination of hardware, software, and a Message Queuing Telemetry Transport (MQTT) protocol is presented.*

Key Words: SCADA, sensors, analytics, grid

1. INTRODUCTION

Autonomous systems are increasing in deployment and use within the electric power grid. The typical situation is for sensors reporting their measurements to the overarching Supervisory Control and Data Acquisition (SCADA) system. When coupled with an array of associated applications, such as an electric utility's Outage Management or Servicing Dispatch, the SCADA system is frequently able to identify the grid network "problem" within a small region (single or few km distance). In the case of an electrical outage, the utility will still dispatch a service vehicle (utility truck) with an associated cost of approximately $1000 USD. An alternative situation is to use a sensor laden drone which is sent to the vacinity of the outage, locate the outage and report to SCADA of its "findings". The overall project described in this paper involves the development of such a system capable of autonomous inspection and reporting for improved grid reliability. The system utilizes advanced sensors, generative design AI, communications, autonomic systems, swarming technologies, energy harvesting, collaborative operation and training of grid state measurements. The initially envisioned operational scenario is depicted in Figure 1. Table 1 lists parameters to be measured for operation of the electric grid.

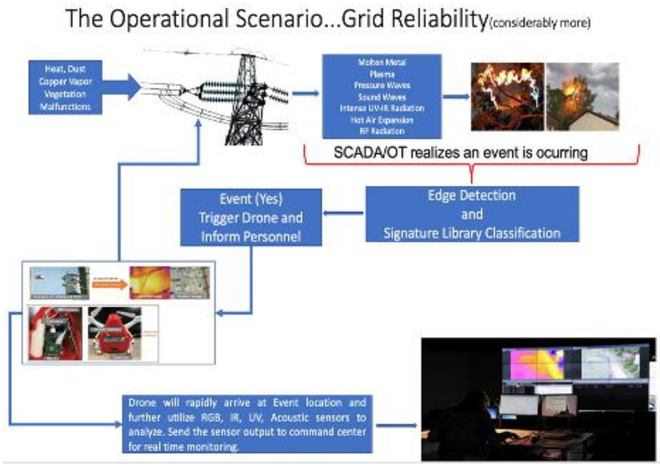

Figure 1. A simplistic flow-chart illustrating the steps in system performance.

With this goal of having an AI engine working in concert with a utility's SCADA system and a robotic control system (RCS), the issues associated with bi-directional information flow arise. The operational scenario of integrating SCADA with an RCS, Figure 2, crosses into the realm of sensor/data fusion [1].

States/Parameters	Directly measured or calculated from measurements	Sensors/meters required	Description and Note
Visual inspection	Direct observation	Photography, video monitoring, ultraviolet imaging	Deployed by drones, robots, manual
Tempera-ture	Measured	Thermocouples and other point temperature sensors, IR imaging techniques	Point sensors identify single temperature locations while imaging tools such as IR imaging can also identify component hot spots while quantifying local temperature
Chemical analysis	Measured	DGA	Presence of N_2, O_2, H_2, CO_2, CO, CH_4, H_2, C_2H_6, C_2H_4, C_2H_2 in transformer insulating liquid or above the transformer oil in the gas phase
Tension	Calculated and Measured	Strain sensor, level or height monitoring	Can be inferred from direct strain sensors or indirect measurements such as line sag for transmission lines
Motion	Calculated	Vibration sensor, strain sensor, imaging based techniques	Camera imaging based methods with associated data analytics and motion proxies such as vibration or strain sensors.
Electrical equipment parameters	Calculated and Measured	Voltage and Current Transducers	Volt, current, phase angle, see previous sections on power flow and electrical grid state
Electrical discharge and corona	Calculated and Measured	Direct or calculated leakage current, local RF and static electric fields	Important for medium and high voltage energized assets including transformer bushings, transmission lines, etc.
Load tap changer position	Measured	Mechanical relay switch position	
Insulation oil level monitoring	Measured	Direct imaging or level indicator measurements	

Table 1. Key States and parameters relevant for electrical system asset monitoring and fault diagnosis.

Note that in order to achieve the measurements outlined in Figure 2, a wide variety of types of sensors are required. In addition, the communication and interfacing protocols and bandwidths involved may require various intertwined architectures. Of particular note is that the utility where the performance validation testing is occurring, Electric Power Board (EPB, Chattaooga TN USA), uses the Distributed Network Protocol 3 (DNP3) [2] in its large scale SCADA system[1].

Field testing is conducted at EPB's utility training site, shown in Figure 2. A Wi-Fi "umbrella" covers the site thereby providing bi-directional, IP addressable, communications for drone-based sensors command and control. Separately - and initially - control of the drone itself is performed via manual non-WiFi communications.

Figure 2. Aerial view of EPB Training Site.

The system design relies on a second SCADA system using Ignition 8.1[2], an integrated software platform for SCADA systems. Via this design, SCADA-B collects data from a variety of drone-based sensors. SCADA B performs data registry and minor sensor signal processing then passes the measurements to SCADA-A. A representative image of SCADA-B's display panel is presented as Figure 3.

Figure 3. SCADA B dashboard display system (representative).

A classic issue with SCADA is the selection of the data protocol. The MQTT protocol [3] was chosen for use with a Raspberry Pi Zero microcontroller acting as the information broker. Companion with the broker is the need for clients. An ESP8266 WiFi development board was selected to serve as the publisher client and the MQTT Engine client (Ignition 8.1) as the subscriber client.

2. METHOD

Hardware and software details associated with the protocol play a fundamental role in designing an overall drone-based sensors and communications system for use in a battery operated, low power consumption scheme. The MQTT protocol is ideal for such a situation of connecting devices with low internal memory and processing power, such as temperature and humidity sensors. MQTT allows bi-directional communications between the core elements by applying a publish/subscribe architecture, thereby increasing information transport transparency.

Fundamentally, an MQTT architecture consists of two systems: clients and brokers. In order to receive specific data from a publisher, a client must subscribe to the data's topic. The MQTT broker acts as the "central hub" of the system, enabling this publish/subscribe (pub/sub) communication system. This situation is presented as Figure 4.

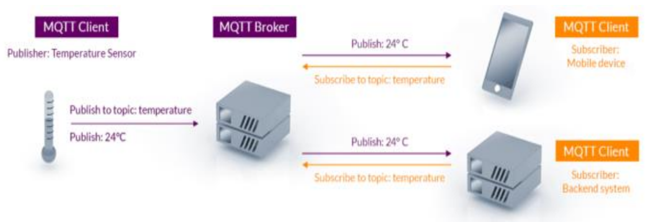

Figure 4. MQTT architecture [4].

A key challenge associated with new and emerging sensing and measurement technologies required for asset monitoring applications is the need for compatibility with electrically energized components. In the case of applications within the distribution system,

[1] Referred to as SCADA-A.

[2] Referred to as SCADA-B.

cost is also a key factor that will drive new technology development for new lower cost sensing solutions.

2.A. The Publisher Client

With the publisher client being responsible for publishing data from a variety of sensors placed on the drones, timing, synchronization, and alerting of "publishing" each sensor reading is important. This is acheived, by having each time a sensor takes a reading, that reading is stored in the sensor's topic and published by the ESP8266 WiFi development board. This was accomplished by utilizing the ESP8266Wifi library and the PubSubClient library found in the Arduino IDE[3]. Instead of using multiple parallel boards to support the transport of measurements from several sensors, a 4051 multiplexer was used. This board provided the capability to connect to up to eight analog sensors at a time. This multiplexing technique minimized the space required to interface the sensors with the drone.

2.B. The Broker

Drones have limited carrying capacity, both in terms of weight and size. Because of these restrictions, it was decided to use a Raspberry Pi Zero as the system microcontroller due to its minimal hardware footprint. This microcontroller has low power consumption yet provides acceptable computational capacity. From a software perspective, a lightweight open-source broker entitled "Eclipse Mosquitto" was installed onto the board. This hardware+software combination of the Raspberry Pi Zero and Eclipse Mosquitto provided the functionality to subscribe to topics sent from the ESP8266 as well as publish them to other clients.

2.C. The Subscriber Client

The MQTT Engine client (Ignition 8.1) acts as the subscriber client in this SCADA system. Ignition 8.1 has worked alongside Cirrus Link Solutions to offer the Cirrus Link MQTT Engine Module in order to make the implementation of MQTT protocols more accessible [5]. this module was used to configure the MQTT Engine client, connect it to the broker, and subscribe to all the topics being published. After a topic is received, Ignition 8.1 stores it as a custom tag. As the MQTT Engine client receives a sensor reading inside a topic, the value of the topic's respective tag updates. The entire system is represented in Figure 5.

[3]An IDE (Integrated Development Environment) is a software that facilitates the development of applications [5].

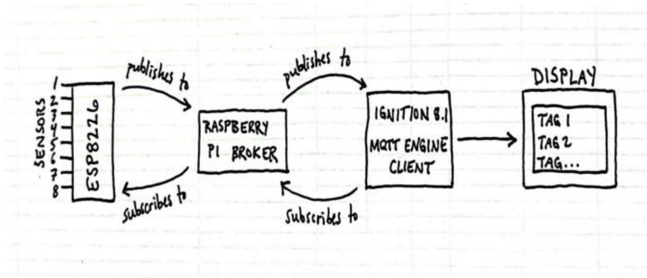

Figure 5. Flowchart of the MQTT system.

3. DEMONSTRATION

In order to demonstrate the functionality of this SCADA B system, a graphical user interface (GUI) was developed using Ignition's Designer. This utilizes a tag from each drone-based sensor. The drone makes measurements from sensors that are trained on, in the case of that shown in Figure 5, an electrical transfomer.

Figure 5. Drone inspecting a distribution transformer.

Every second, the tag updates and the temperature value changes. Correspondingly, the GUI's temperature indicator - along with the drone's position - is updated on the gauge, Figure 6.

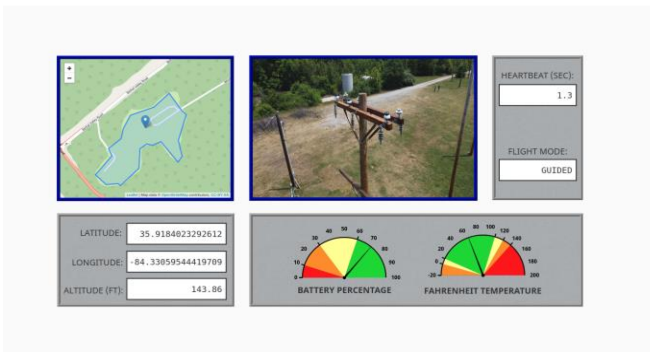

Figure 6. The SCADA-B system's graphical user interface.

4. CONCLUSIONS

A functional SCADA system using various sensors, an ESP8266 board, a Raspberry Pi Zero, and Ignition 8.1 has been designed and implemented. It was determined that the MQTT messaging protocol was the most effective means of data communication for this project due to its strong relationship with low power devices. An ESP8266 WiFi development board was selected and implemented as the system's publisher client, a Raspberry Pi Zero served as the broker, and Ignition 8.1's MQTT Engine client was the subscriber client. Lastly, a graphical user interface was designed and implemented to confirm that the sensor data is being received and updated continuously.

Moving forward, it is suggested that a stable line of communication between Ignition 8.1 and the drone's ground station application, Mission Planner, be developed. Ignition 8.1 has a multitude of applications that allow bi-directional communication through the software, meaning it could send commands to the project's programmable logic controllers (PLCs) based on the sensor data. Integration with additional systems - following the Industry 4.0 demonstration application [6] - will link developments within the electric grid and UAS research areas.

5. REFERENCES

1. Corder, J. A. (1996). *New World Vistas: Air and Space Power for the 21st Century*. USAF Scientific Advisory Board.

2. https://en.wikipedia.org/wiki/DNP3

3. Koelsch, J. (2022, February 25). *Why the MQTT Protocol is So Popular*. Automation World. https://www.automationworld.com/factory/iiot/article/22080946/why-the-mqtt-protocol-is-so-popular

4. *The Standard for IOT Messaging*. MQTT. (n.d.). https://mqtt.org/

5. Gillis, A., & Silverthorne, V. (2018, September 6). *Integrated Development Environment (IDE)*. Software Quality. https://www.techtarget.com/searchsoftwarequality/definition/integrated-development-environment

6. Reynolds, W. (2020, March 18). *Industry 4.0 Demo Application*. LinkedIn. https://www.linkedin.com/pulse/industry-40-demo-application-walker-reynolds