Wireless Water Flow Monitoring System

Bilal Ayyache
School of Engineering
University of Guelph
Guelph, Canada
bayyache@uoguelph.ca

Harmeet Singh
School of Engineering
University of Guelph
Guelph, Canada
hsingh25@uoguelph.ca

Neel Bhandari
School of Engineering
University of Guelph
Guelph, Canada
nbhandar@uoguelph.ca

***Abstract*—Wireless systems are used to automate tasks in industrial settings and collect data from environment. In this paper, a wireless flow monitoring system is designed using a micro-controller, and a micro-computer. The main purpose of this system is to collect water-flow data from pipes and provide this data in a presentable fashion on a dashboard. The data created was presented using a litres-per-minute vs time graph, and the user was given the option to receive an alert when consumption level reaches a finite limit set by the user. The network created uses a star topology network where sensors in-site collect data and send it to the main base. The main base of the system is an MQTT server that runs on a Raspberry Pi. The sensors utilize a NodeMCU and uses WIFI module to send data of water flow to the MQTT Server. The dashboard is created using NodeRED which subscribes to the MQTT server and listen to the messages received. The system designed is able to provide the client with water consumption data that can help them utilize usage, reduce water consumption, and increase profit earned. The results of the system created was accurate and precise as the sensors accuracy was high. In this report the design methodology, related work, hardware, and the software implementation was described.**

***Index Terms*—MQTT, NodeRED, Wireless network, Star topology, NodeMCU, Raspberry Pi, Water flow sensor, Wifi, Database, Arduino**

I. Introduction

Following the bill legalizing recreational cannabis consumption, there has been a massive growth in the cannabis production industry. However, due to its novelty the amount of water required to obtain the greatest cannabis yield is unknown. As a result, this contributes to a wastage of water or insufficient cannabis production for the producers. To solve this issue, we have suggested the use of a wireless sensor network. This network will monitor the water flow rate at each node of the water irrigation system. Specifically, in this project we will be collecting water flow information from 10 different growing rooms in a dispensary. The dispensary is located in Winnipeg, Manitoba under the name of Canendo Cannabis. Water flow data will be collected from a one-inch pipe using a water flow sensor and sent to an MQTT broker using a NodeMCU as a micro-controller. The data is filtered, processed and uploaded to the server in real-time. Data sent in to the MQTT server is managed using a raspberry pi along with node red. The raspberry pi is the base of the network. Data from all nodes is collected and stored in an organised manner. This data is later uploaded to a mongoDB database for future implementations such as a custom Dashboard. By monitoring the water flow rate in different growing rooms, companies like Canendo Cannabis will be able to monitor exactly how much water flow rate is required to have the ideal yield. Ultimately monitoring the water flow rate will allow Canendo Cannabis to maximize their yield, allowing them to increase profits.

Figure 1 illustrates an overview of what the network system looks like. A water source with a pressure of 30 PSI enters 10 different rooms through a 1 inch pipe. Each room will have a dedicated system that will upload the water flow usage to the MQTT Broker. The Rasberry Pi will capture these messages from the broker and provide useful data to the client. Data includes historical and real time water consumption graphs.

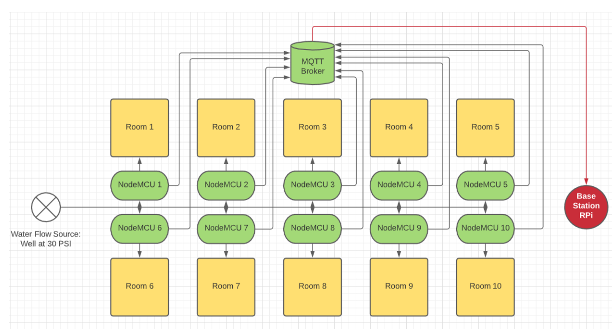

Fig. 1. System Overview

II. Related Work

Currently, StreamLab has a Non-Invasive Water Flow Monitoring system to track water usage in residential homes. Their system uses one device to monitor the entire house using no additional nodes. Instead, they use an ultrasonic sensor to measure and detect the changes in flow rate in the house. Stream labs system is wifi enabled, allowing them to not require a central hub. [13] Our rendition of a water flow monitor system differs drastically from stream labs. Our system is very flexible, as it uses multiple sensors scattered around in the facilities to get the water flow rate in a specific room. In our system, a raspberry pi was used as our central hub. This is beneficial as it allows us to store data for an extended amount of time on an SD card attached to the raspberry pi. Having more data is very useful for business owners as it allows them to review and analyze more material.

Thus, this gives them a more accurate representation of what is actually happening.

Our system is better suited for a commercial water flow monitoring system as it is easily scalable.When trying to get the flow rate of water entering into one additional room all our system would require is one more node consisting of a sensor and a NodeMCU whereas if using a Streamslabs system an entirely new device would need to be purchased and installed on the pipe that is bringing water into the room.

Fig. 2. Non-Invasive Water Flow Monitor By StreamLabs

Flowie is Alert Labs new water flow sensor, it is used to monitor water usage as well as leakage in both residential and commercial applications. This system uses one unit, it must be attached to the water meter and requires a cellular connection to communicate data back to the dashboard [15]. Our system differs from Alert labs system in many ways such as; type of sensor, the number of sensors a system can handle and the communication method. Alert labs system uses ultrasonic sensors and our system uses a water flow sensor, the advantages of using the water flow sensors is the increased accuracy of the measured flow as well as the affordability [16]. Flowie is a single node system, whereas our system uses multiple nodes. As Alert Lab only has one node in their system it is unable to get the water usage at every individual room, rather it looks at the home or commercial unit as a whole and is only able to tell the water flow and water usage of the entire facility. Our system uses multiple sensors, scattered around the facility to measure the water flow rate entering every room. Lastly, as Flowie uses a cellular connections it can become extremely cost inefficient for users, cellular data is typically bought in a certain amount(5 GB,10 GB) whereas WIFI is generally unlimited, thus putting a limit to the amount of data that can be communicated back to the dashboard when using a cellular connections. If the limitation were to be removed and all the data was sent to the dashboard the cost of the cellular connection would be substantially larger than the cost of WIFI [17]. As our system is both more cost efficient and measures more reliable values for water flow rates throughout facilities,therefore making our system is better suited for both residential and commercial applications.

Fig. 3. Flowie By Alert Labs

III. System Architecture

A. Hardware Interface

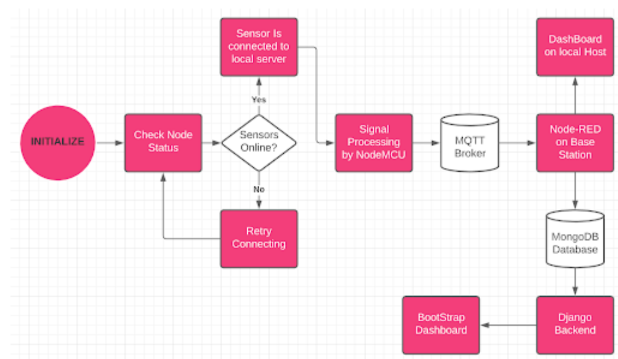

Fig. 4. Hardware Flow

The ”DIGITEN G1” Water Flow Hall Effect Sensor Switch Flow Meter Flowmeter Counter 1-60L/min” was used to output digital values in order to successfully estimate the water flow through a pipe. This sensor was chosen as it checked the criteria requirements imposed by the client and helped reduce cost. The criteria requirements were; the water flow rate must be able to measure from 1-60L/min and must be able to withhold the water pressure of up to 30psi(0.2Mpa approximately). NodeMCU attached to the sensor is used as a micro-controller to process and send the values to mosquitto MQTT Broker. NodeMCU was chosen as it makes use of ESP8266 which allows us to connect to wifi, save energy, and supports the MQTT IoT protocol [23]. Raspberry Pi Model 3 is used as a microcomputer that runs a Node-RED hardware flow. Node-RED is used for wiring together the hardware devices, APIs and online services as part of the IoT [25]. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. The runtime is built on Node.js.

The flows created in Node-RED are stored using JSON. JSON format was preferred over the XML format as its easier to control and manage data using JSON.

B. Software Interface

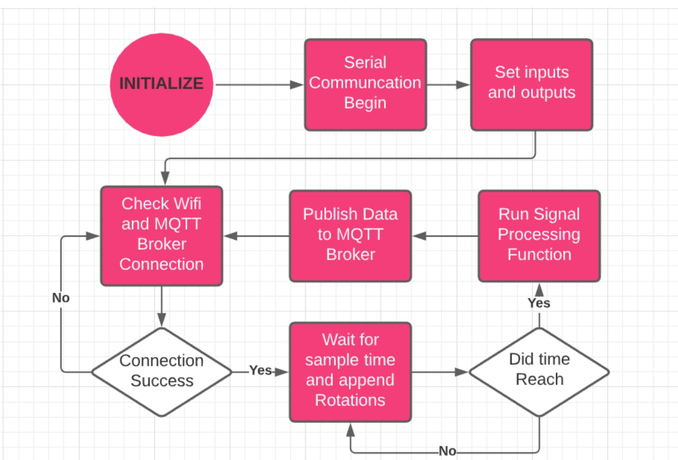

Fig. 5. Software Flow

The software application that runs on the NodeMCU microcontroller is described using the software flow diagram in figure 5. The software application first initializes the input and the output of the system. Serial communication is started and the inputs are defined. The wifi connection is checked, if the NodeMCU connects to the wifi it tries to connect to the MQTT server. If the NodeMCU fails to connect to wifi, a message will be sent to the user and will retry connection. If connection to MQTT server is successful append the rotations in the sensor and wait until the sample time is over. After the sample time is over pass in the number of rotations in sensor to the signal processing function and calculate the litres per sample time. After the litres/min calculations are complete, publish the data to the MQTT server and run repeat. This software application runs forever to provide the user with real time data.

The Software code to process the data provided by the sensor is presented in figure 6:

```
if (now_sensors - CountStart > update_time_sensors)
{
  CountStart = now_sensors;
  Serial.println("Resume--------------------------------------");
  if (CountFlow > 1500000 || CountFlow == 0) {
    clientMqtt.publish(mqtt_pub_topic_waterflow, "0");
  } else {
    //Flow pulse characteristics (6.6*L/Min)
    CountLitre = (CountFlow*6.6/2815562);
    snprintf (msg, 10, "%6.2f", CountLitre);
    clientMqtt.publish(mqtt_pub_topic_waterflow, msg);
    Serial.print(msg);
    Serial.println(" liters");
  }
  Serial.println(CountFlow);
  Serial.println("");
  CountFlow = 0;
}
```

Fig. 6. Software Code for signal processing

C. Sensor Unit

Water flow sensors are installed to accurately measure and calculate the amount of water flowing through the pipe or a water source. The unit used is liters per minute and provides us with data containing the amount of water flowing through a water source or pipe in the given unit. The water flow sensor used consists of a plastic valve and a water rotor, the water is passed through the valve, from where it rotates the rotor. The speed of the rotor is observed and is used to calculate the output as a pulse signal. The sensor makes use of a hall effect sensor to help and measure the water flow. The hall effect sensor calculates the output, and the output is then returned as a pulse signal. The hall effect sensor makes use of the Hall Effect principle, according to which, a voltage difference is induced in the conductor due to the rotation of the rotor [22]. DIGITEN G1" Water flow hall effect sensor was used for this system. A few other competitors and water flow sensors were also considered before finally selecting the DIGITEN G1 sensor. Psyelec 1Pcs G3/4" Hall Effect Liquid water flow sensor is made out of brass and holds well under extreme conditions however the flow range is 1-30L/min and this does not fit our criteria as our client wants to be able to measure the flow rate ranging from 1-60L/min. Another contender was the DIGITEN G/4" water flow hall sensor and this sensor also makes use of the hall effect principle [24]. Although, due its expensive price this sensor was not chosen. The Digiten G1 water flow hall effect sensor was chosen due to its ability to withstand water pressure of upto 1.75 Mpa, generate water flow data ranging from 1-60L/min, and to maximize profits and reduce cost.

Fig. 7. Use Case Diagram For Testing

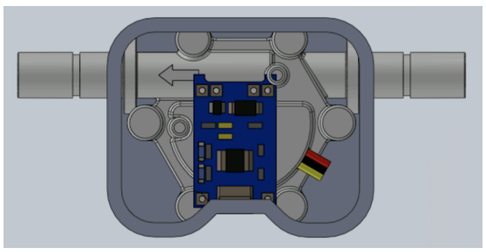

Fig. 8. 3D Model Of System

D. *Alternative Design*

A different design was considered, in this design the Raspberry pi was removed and data was taken and sent straight from the NodeMCU to the database. However, to implement this design Google Firebase had to be implemented. It was later found that this data is harder to manage and memory space is limited if Firebase was used.Hence, due to the flexibility achieved while designing the dashboard using a Raspberry Pi, allowing us to create a more professional looking application.

Another alteration that could have been made to the design was replacing the NodeMCUs with a microcomputers. However, this approach was found to be expensive and contributed no additional value to the design. The system does not require the computational power of a microcomputer, the computational power provided by the microcontroller seems to be sufficient and helps to reduce cost, while also requiring less energy [21]. Hence, it was decided that replacing the NodeMCU's would be excessive.

IV. Communication

There are various wireless approaches that can be used to design a wireless network. The 2 main approaches are the direct communication model and the multi-hop communication model. In the direct communication model all nodes communicate directly to the base, whereas in the multi-hop communication model nodes act as relays and forward data from one node to another to ultimately reach its destination. Although both approaches are good they come with their own advantages and disadvantages. As such, we have employed the direct communication model approach. In our case all the nodes are the sensors with the node MCU's attached to them and they all communicate directly to the Raspberry pi. We chose to stick to a direct communication model as they are less expensive and simpler to implement. As the data is only sent to the base station rather than to several other nodes ,the probability for losing data decreases. The security of the network increases when using a direct communication model as the node endpoints work independently, therefore if one node was to be attacked the remainder of the network could still work properly. The main disadvantage of the direct communications approach is the limited transmission range but as our nodes all send data to a local host [14], in our case Raspberry pi, allowing us to not need to have a large transmission range.

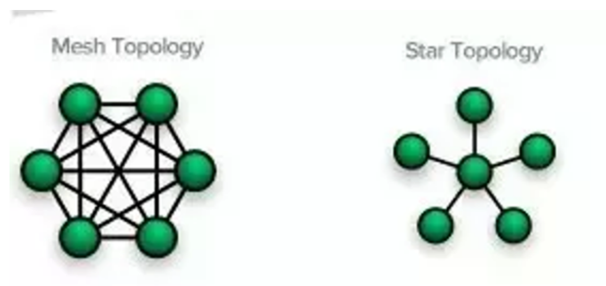

Fig. 9. Network Topology

Using the MQTT protocol allows for fast and easy communication between nodes and the data base [18]. Using Publisher/ Subscriber, an asynchronous messaging service, data is taken from the raspberry pi using NodeRED and is sent to the MQTT server. As Publisher/ Subscriber offer both message storage and real time message delivery [19] it is ideal for our IoT application. Unlike traditional message queuing pub/sub allows for multiple consumers to receive each message and ensures messages are received in the order in which they were produced [20].

V. Data Analysis

The MQTT server is a broker which gives access to create a subscription to publish. Using this broker values are read from the server and stored in JSON format. Node-RED is used to set modules to read the JSON files from the MQTT servers. The broker provides functionality to collect the data and store it at a single location, making it easier for the Node-RED to handle this data. The MQTT server works on port 1883 and the NodeMCU works on port 1880. As the MQTT server and the NodeMCU work on different ports, data is able to be transmitted and received without losing any information. The MQTT server sends data to port 1883 using Node-RED, this data is read from port 1880 and sent to the database. Once the data from all the sensors is collected on to the database, we will no longer need to use the MQTT server or Node-RED. All the data collected is published to the database in real time, giving it the ability to produce most accurate readings for the water flow at any given time.

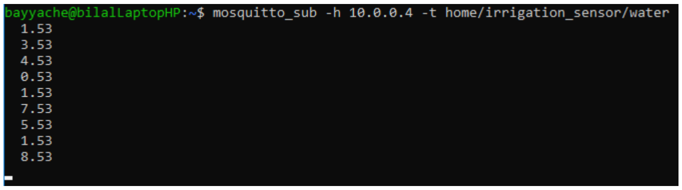

Fig. 10. MQTT publisher

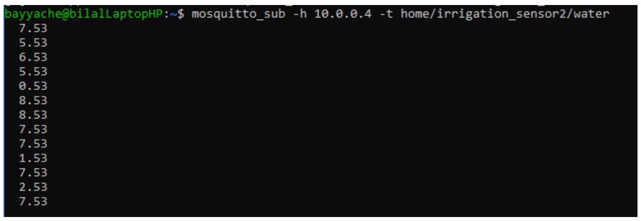

Fig. 11. MQTT Subscriber

Figure 10 and 11 monitors the MQTT server through subscribing to the local host on port 1883. Figure 10 presents the data from the first water sensor in real-time. Every 5 seconds, the sensor sends the L/min of water-flow from the first pipe. Using the command function presented in these figures, the data can be monitored from any computer connected to the same network as the base station.

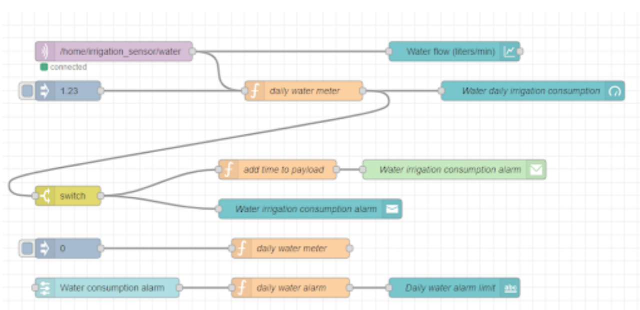

Fig. 12. NodeRED Server

Figure 12 describes the NodeRED server on the base station. The NodeRED application subscribes to the first and second sensors' MQTT server. The data is extracted from the MQTT server and used to present the data on a graph. The daily amount of water is calculated and displayed to the user using a gadget. If the daily water meter value exceeds the value set by the user, an email is sent to the user and a notification is pushed to the dashboard.

VI. RESULTS

Results are presented in figure 13 and figure 17. In figure 13 The water was turned off for a very long duration. After the long duration of a constant flow, the water flow increased and fluctuated. The dashboard created allows the user to send an alert to an email if water flow exceeds a limit set by the user. The dashboard presents real-time data and stores historical data up to 3 days. The purpose of this figure is to present the functionality of the system when water flow goes from zero to a measured value. The figure below presents a graph of litres per minute vs time:

Fig. 13. Results Graph 1

Figure 17 presents the water flow data in figure13 zoomed in. In this experiment the water flow was altered by increasing and decreasing the flow to make sure that the meter presents accurate information. An alarm was sent as the water flow meter exceeded 8 Litres per minute. The Data is sent to the server every 5 seconds creating a digital graph. From the results, it was concluded that the sensors computes accurate information about the water flow. The information was verified using a calculated amount of water which was passed through a pipe in which the sensor was connected to.

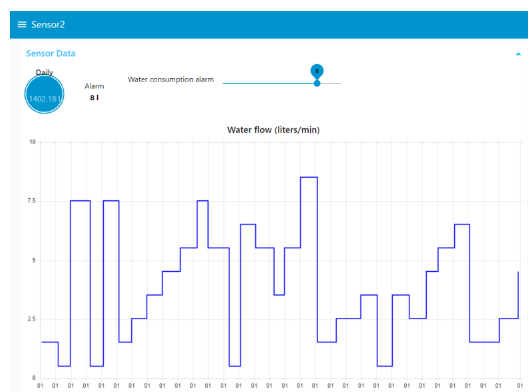

Fig. 14. Results Graph 2

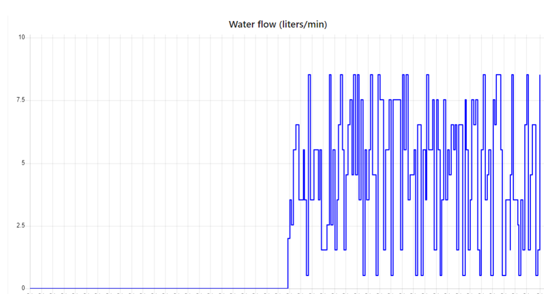

Fig. 15. Results Graph 3

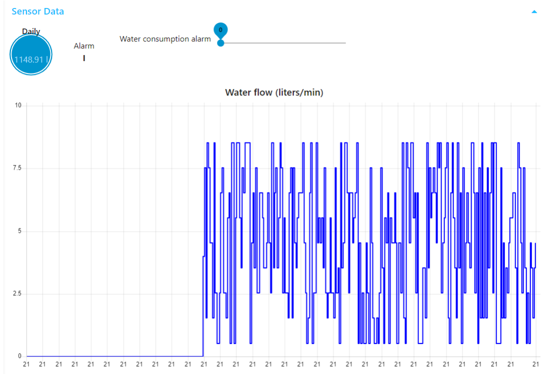

Fig. 16. Results Graph 4

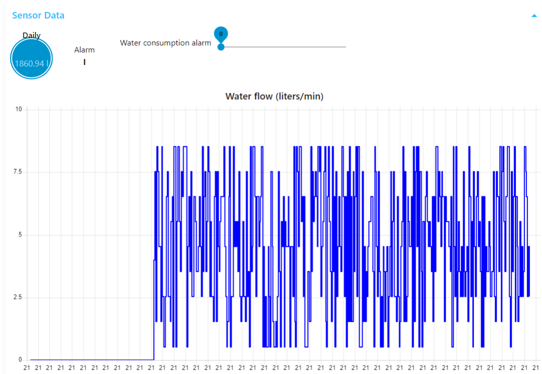

Fig. 17. Results Graph 5

VII. Conclusion

In conclusion, a wireless water flow monitoring system was designed to collect the water-flow data from pipes and provide this data in a presentable fashion on a dashboard. This network was created using multiple water flow sensors paired with a microcontroller and one microcomputer. The nodes in the system used NodeMCU along with WiFi to send the data collect to the MQTT server. The network uses a star topology to allow for secure and fast data collection. The central hub for the system is the MQTT server that is running on the raspberry pi microcomputer. Through the use of NodeRED a dashboard was created to professionally display the data collected from the sensors. As our system is cost efficient, accurate and scalable it is the perfect candidate for water flow monitoring system in an industrial application.

References

[1] Prentice Hall, Digital Process Control: Analysis and Design [Textbook], 2014.

[2] Ziegler JG, Nichols NB. Optimum settings for automatic controllers. trans. ASME. 1942 Nov;64(11).

[3] M. Faheem, V.C. Gungor Computer Standards & Interfaces Capacity and spectrum-aware communication framework for wireless sensor network-based smart grid applications Comput. Stand. Interfac., 53 (March) (2017), pp. 48-58

[4] A. Helmy The ACQUIRE mechanism for effcient querying in sensor networks Electr. Eng. (2003), pp. 149-155

[5] Y.M. Hasn, S.A. Hussein Energy efficient routing protocols for wireless sensor networks abstract: designing an energy efficient routing protocol for wireless sensor network (WSN) has for IOSR J. Electron. Commun. Eng., 11 (2) (2016), pp. 2278-2834

[6] M.M. Warrier, A. Kumar An energy efficient approach for routing in wireless sensor networks Procedia Technol., 25 (Raerest) (2016), pp. 520-527

[7] L. Subramanian, R.H. Katz An architecture for building self-configurable systems 2000 1st Annu. Work. Mob. Ad Hoc Netw. Comput. MobiHOC 2000 (2000), pp. 63-73

[8] H. Lin, P. Chen, L. Wang Fan-shaped clustering for large-scale sensor networks Proc. - 2014 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2014 (2014), pp. 361-364

[9] Y. Kiri, M. Sugano, M. Murata Self-organized data-gathering scheme for multi-sink sensor networks inspired by swarm intelligence First Int. Conf. Self-Adaptive Self-Organizing Syst. SASO 2007 (2007), pp. 161-170

[10] F.B. Tiago Camilo, Carlos Carreto, Jorge Sa Silva An energy-efficient ant-based routing algorithm for wireless sensor networks IEEE Sens. J., 16 (20) (2016), pp. 7393-7400

[11] T. White, B. Pagurek, F. Oppacher Connection management using adaptive agents Proceedings of 1998 international conference on parallel and distributed processing techniques and applications (PDPTA '98), pp. 802-809, CSREA Press (1998)

[12] K.T.K. Tian, B.Z.B. Zhang, K.H.K. Huang, J.M.J. Ma Data gathering protocols for wireless sensor networks with mobile sinks GLOBECOM 2010, 2010 IEEE Glob. Telecommun. Conf. (2010)

[13] "Non-Invasive Water Flow Monitor," Water Security Solutions, 29-Aug-2019. [Online]. Available: https://wssus.com/product/non-invasive-water-flow-monitor/. [Accessed: 06-Dec-2020].

[14] P. on D. 1, P. on N. 24, P. on N. 17, P. on N. 10, and P. on N. 3, "Mesh vs Star Topology - Find your right IoT Architecture: BehrTech Blog," BehrTech, 28-Oct-2020. [Online]. Available: https://behrtech.com/blog/mesh-vs-star-topology/. [Accessed: 06-Dec-2020].

[15] Alert Labs, "Flowie Water Sensor," Alert Labs. [Online]. Available: https://alertlabs.com/products/flowie-water-sensor. [Accessed: 06-Dec-2020].

[16] K. Gross, "Ultrasonic Sensors: Advantages and Limitations," MaxBotix Inc., 28-Oct-2020. [Online]. Available: https://www.maxbotix.com/articles/advantages-limitations-ultrasonic-sensors.htm/. [Accessed: 06-Dec-2020].

[17] Connectify, "What is the difference between Wi-Fi and Cellular connections?," Speedify Knowledge Base. [Online]. Available: https://support.speedify.com/article/257-difference-between-wifi-cellular-connection-avoid-connection-interruptions. [Accessed: 06-Dec-2020].

[18] J. MSV, "Get to Know MQTT: The Messaging Protocol for the Internet of Things," The New Stack, 28-Feb-2019. [Online]. Available: https://thenewstack.io/mqtt-protocol-iot/. [Accessed: 07-Dec-2020].

[19] "What Is Pub/Sub? — Cloud Pub/Sub Documentation — Google Cloud," Google. [Online]. Available: https://cloud.google.com/pubsub/docs/overview. [Accessed: 07-Dec-2020].

[20] J. Bock, "Comparing Publish-Subscribe Messaging and Message Queuing - DZone Integration," dzone.com, 15-Apr-2019. [Online]. Available: https://dzone.com/articles/comparing-publish-subscribe-messaging-and-message: :text=Like message queuing, publish-subscribe,each message in a topic.text=Publish-subscribe messaging example. [Accessed: 07-Dec-2020].

[21] palda himaja 4. September 2020 at 09:10, Bilal Malik 7. September 2020 at 16:12, Ashish Shukla 16. September 2020 at 15:41, ben 22. September 2020 at 03:52, Kees Baltus 2. October 2020 at 13:36, and elle 11. October 2020 at 10:03, "Arduino vs ESP8266 vs ESP32 Microcontroller Comparison," DIYI0T, 01-Nov-2020. [Online]. Available: https://diyi0t.com/technical-datasheet-microcontroller-comparison/. [Accessed: 07-Dec-2020].

[22] N. *, "Water Flow Sensor - Working Principle , Applications Examples.," ElProCus, 17-Aug-2019. [Online]. Available: https://www.elprocus.com/a-memoir-on-water-flow-sensor/. [Accessed: 07-Dec-2020].

[23] Lee, L. A. veteran programmer, Lee, A. veteran programmer, E. F. says: B. says: L. says: M. 26, and L. says: "Introduction to IoT: Build an MQTT Server Using Raspberry Pi," App Code Labs, 10-May-2019. [Online]. Available: https://appcodelabs.com/introduction-to-iot-build-an-mqtt-server-using-raspberry-pi. [Accessed: 07-Dec-2020].

[24] G3/4 Water Flow sensor. [Online]. Available: https://www.openhacks.com/uploadsproductos/g3_4_water_flow_sensor_-_wiki.pdf.

[25] "RED," Node. [Online]. Available: https://nodered.org/. [Accessed: 08-Dec-2020].