A Cyber-Physical System for Remote Monitoring and Control of Water Purification and Recycling Systems

Niteesh S Shanbog
Societal Electronics Group
CSIR Central Electronics
Engineering Research Institute
Pilani, India
nshanbog.1@gmail.com

Priyansh Parikh
Electrical and Electronics Engg.
Birla Institute of Technology
and Science Pilani
Goa, India
priyanshparikh1211@gmail.com

Shubham Gelera
Electronics and Intrementation Engg.
Birla Institute of Technology
and Science Pilani
Pilani, India
gelerashubham@gmail.com

B. A. Botre
Societal Electronics Group
CSIR Central Electronics
Engineering Research Institute
Pilani, India
bhau@ceeri.res.in

Anil Kumar Saini
Societal Electronics Group
CSIR Central Electronics
Engineering Research Institute
Pilani, India
aksaini@ceeri.res.in

***Abstract*—Water Purification and Recycling (WPR) plants are generally controlled using expensive PLC-based systems. This paper proposes a general architecture for building an integrated Cyber-physical System (CPS) incorporating IoT techniques for remote monitoring and control. The developed system includes a provision to interface a wide variety of water quality sensors like conductivity and pH as well as sensors to monitor the status of the WPR plant. Remote monitoring of the WPR plant is achieved by ingesting the sensor data into a cloud-based IoT platform. The developed system is also equipped with a touchscreen Human Machine Interface for the display of plant parameters as well as an SD card module for local storage of data. Low-cost microcontrollers are used to design the Electronic Monitoring and Control System and the IoT gateway. Kaa, a cloud-based IoT platform built using open source components has been chosen to host the sensor data for visualisation and further analytics. This system can provide a general framework to design a low-cost IoT solution for monitoring and controlling a Water Purification and Recycling plant.**

***Index Terms*—IoT, water quality monitoring, remote monitoring, MQTT, IoT platforms, cyber-physical systems**

I. Introduction

The importance of clean drinking water cannot be understated. Access to safe, sufficient and affordable water has been known to be a major facilitator in economic development and poverty reduction. While building the infrastructure for recycling water is one challenge, monitoring the water quality and ensuring that treated water is in accordance with the established standards is another challenge. Traditional methods of water quality monitoring involve collecting samples and manually performing chemical analysis in labs. The time component in this method is however huge and in cases where pathogenic water is allowed to pass into domestic consumption, the results might quite well be disastrous. Advances in sensor technology, computing technology and communication methods have made possible real-time data collection, on-site data processing and remote monitoring and control [1]. The cost of these specialised sensors and powerful nodes was still prohibitive till the advent of IoT technology. Low-cost, low energy, edge computing devices deployed with portable sensors and the internet can be deployed to monitor water quality remotely from any corner of the world. These Cyber-physical Systems (CPS) strive to automate physical processes by incorporating aspects of control, communication and computing [2].

Sensors for water quality monitoring have also followed a similar evolutionary trend enabling water quality monitoring to move from the traditional Manual Lab-Based to the more recent Wireless sensor node solutions [3]. Portable and low-cost sensors with electronic transducers along with the capability to transmit the stimuli reading either through wired or wireless methods of transmission are being used in most IoT based water quality monitoring systems. A simple low-cost design of various water quality sensors has been shown in [4] along with a ZigBee based notification system. Even though this system has been implemented on a stationary sample of water, the interfacing of the sensors to the microcontroller with signal conditioning offers a good starting point for anyone looking to build a water quality monitoring system from scratch.

A low-cost method to remotely monitor and control the operation of a wastewater treatment plant has been demonstrated in [5]. Temperature, flow rate and water level of the wastewater is measured in different stages of the wastewater treatment plant by a low cost ESP8266 NodeMCU microcontroller. An

IoT gateway is created with the ESP8266 as an MQTT client which publishes the sensor data into the Losant Cloud platform. This prototype provides a low-cost framework on which the entire architecture of an IoT based remote monitoring and control system can be developed, right from the sensors to the cloud-based platform. A similar architecture is used by [6] by using an Arduino microcontroller and MATLAB's ThingSpeak IoT platform. However, this prototype incorporates a wide variety of water quality sensors for measuring the conductivity, colour, Dissolved Oxygen (DO), Ammonia, etc. Another important innovation is the addition of an Artificial Neural Network model to analyse the water treatment pattern and to predict a point when the water becomes untreatable. The implementation of Artificial Intelligence (AI) techniques to give insights on the operation of water treatment plants has been further shown in [7] and [8].

The monitoring and operation of a chlorination based water treatment plant have also been achieved using a similar Arduino based hardware in [9]. In addition to monitoring the chlorine concentration and water levels, the authors propose a Smart Water Management System built on the document-oriented MongoDB database. Data is transferred from the prototype to the custom-built web application using RESTful APIs and the HTTPS protocol. The web application, built of various open-source modules monitors the operation of the treatment plant and is capable of sending alerts using SMS and email.

Various versions of IoT based water quality monitoring system have been developed in [10]–[13] with minor variations in the choice of microcontroller, the types of sensors and the cloud-based IoT platform used. Real-life versions of smart water quality monitoring systems have been shown in [14] and [15]. These implementations form a complete end to end Cyber-physical system, from the sensor networks to the IoT platform for the end-user.

The purpose of this project is to develop an autonomous Cyber-physical System to monitor and control the operation of a Water Purification and Recycling plant. The developments made in this project will set up a foundation to design a general architecture to develop low-cost IoT solutions which can be implemented in small scale WPR plants. The methods to interface various sensors has been provided along with a way to send the data to a cloud based IoT dashboard. Following this section, Section II describes the basic approach used while designing the prototype. Section III describes the physical architecture, the sensor data flows and the operating algorithm used. Section IV shows the simulations and results observed and Section V summarises the work done so far and the conclusions drawn.

II. Approach

Based on the recent developments in remote water quality monitoring in water treatment plants, a low-cost microcontroller-based architecture was developed for interfacing basic sensors like conductivity, pH, level and pressure sensors to a microcontroller and visualising it in a cloud-based IoT platform. Based on the tank levels, control actions have to be released to the various pumps in the plant. A relay circuit was designed for this purpose. Message Queued Telemetry Transport (MQTT) is chosen as the communication protocol for message delivery from the clients to the broker. Compared to HTTP, MQTT sends more data per byte and takes up to 60% less time to send a message [16]. There exist many IoT platforms for sensor data ingestion and analytics. In the present work report, the authors have decided to go with Kaa, an open-source IoT platform to create intuitive real-time dashboards with sensor data. It provides enough flexibility to integrate various third party components like Zookeeper, ElasticSearch and NoSQL to create effective microservices. Instead of connecting the microcontroller directly to the IoT cloud, a Raspberry Pi has been used as an IoT gateway. While this may seem redundant for a single microcontroller, it offers scalability for future installations, as the gateway uses one server connection to connect multiple endpoints to the cloud. The use of an IoT gateway also enables us to choose a base microcontroller with lower computing power.

III. Architecture of Electronic Monitoring and Control System

A microcontroller-based Electronic Monitoring and Control System (EMCS) was designed to be able to collect data from multiple water quality and control parameter sensors and give output signals to control the operation of the water purification and recycling plant. As shown in Fig. 1, the EMCS has a modular architecture making it easy to replace sections of the system for service and repair. The modular design also enables faster and more efficient testing and prototyping. The EMCS is driven by a microcontroller to process the sensor data and generate control signals based on a pre-defined permissive and protection rule set. Conductivity and pH sensors are used to gauge the effectiveness of the purification system and make sure that the treated water is safe for domestic consumption according to standards. Magnetic float level sensors and pressure sensors are used to set alerts about the level of the tanks and measure the pressure in the lines. All the sensor data is processed and calibrated using the equations in Eq. 1 - 4 in the microcontroller and the time series data is stored locally on a micro SD card for further analysis. The data is also sent to a cloud-based web dashboard continuously for remote monitoring and further analytics.

A. Physical Layout

Conductivity, pH and pressure sensors are connected to the microcontroller using analog input pins. The magnetic float level sensors send the level status through a digital ON/OFF signal to the digital pins on the microcontroller. The control signals to the valves and the pumps are given via digital output pins. The AtMega 2560 microcontroller was chosen for the EMCS system due to the availability of sufficient digital and analog pins and interfaces for all the necessary peripherals.

The Real-Time Clock required for the timestamp is achieved using DS3231, an I2C based real-time clock with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. Another alternative to the DS3231 is the much cheaper DS1307. However,it comes with external crystals whose oscillation frequency is easily affected by external temperature. The TCXO in DS3231 is integrated with the IC, along with a temperature sensor. This makes the DS3231 highly resistant to deviations in the external temperature to the standard temperature. The DS3231 RTC module also comes with a CR2032 battery unit which it uses to perform the timekeeping function when the main power to the device is cut off. The chip consumes a minimum of 3μA during the sleep condition making it an excellent low powered modular solution for timekeeping.

A Micro SD card module is also integrated into the microcontroller using SPI mode. The micro-SD card operates on 3.3V. The module has an onboard ultra-low dropout regulator that converts voltage sup to 6V to 3.3V. There's also a 74LVC125A chip on the module for 3.3V - 5V - 3.3V logic level shifting. The constant stream of data generated makes it beneficial to use hardware SPI pins instead of bit-banging techniques as this allows the microcontroller to perform other computational tasks.

A touchscreen LCD is chosen from Nextion's series of Human Machine Interfaces (HMIs). These displays are connecting using the Serial RX and TX pins. The Graphical User Interface (GUI) visible to the end-user is designed using the free Nextion Editor software. The touchscreen LCD is used for local display of plant status and water quality parameters as well as to give emergency stop instructions.

The power supply to the Electronic Monitoring and Control System is designed keeping in mind the power requirements of the microcontroller and its peripherals, the touchscreen LCD as well as communication to the IoT gateway. The EMCS is powered by an Industrial Switched Mode Power Supply (SMPS) which gives an output of 24V from the AC mains supply. The power supply section has two LM2596 adjustable buck converters which can supply up to 3A of current. The first buck converter steps down 24V to 9V to power the microcontroller. Two branches of this line are further passed through two 7805 5V regulators to power the display, RTC and the SD Card modules as well as the analog input section. The second LM2596 buck converter is tuned to convert 24 V to 5 V and powers the optical isolation section provided for the digital outputs. It also takes care of the power requirements for the IoT interface.

It is necessary to divide the main power supply line into separate branches with voltage regulators in each branch as the EMCS system involves multiple peripherals and sensors and might cause voltage drops when they are all connected simultaneously. Since most of the sensors used like conductivity and pH are active sensors, i.e, they require an external voltage or current to produce an output, they are highly susceptible to changes in supply voltage. The current consumption of the touchscreen LCD also varies from 15mA to 510mA

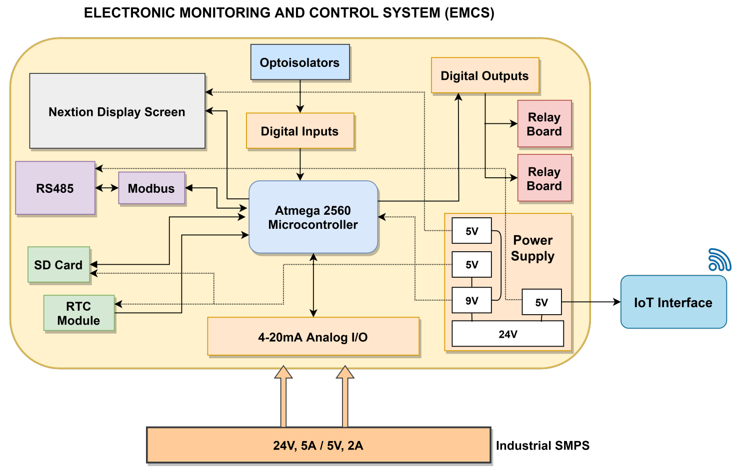

Fig. 1. Physical Architecture of the Electronic Monitoring and Control System

depending on whether the screen is in *SLEEP* mode or awake. Having dedicated power supply branches helps to maintain the integrity of analog sensor measurements as well as ensures the safe operation of the touchscreen LCD.

B. Data Flows

The sensor data flows in the Electronic Monitoring and Control System is shown in 2 The data flows within the EMCS can be divided into two sections: sensor data from the sensors to the microcontroller, and the flow of data from the microcontroller to the cloud-based IoT platform.

1) Sensor to Microcontroller Communication: The conductivity and pH sensors are powered by 230VAC and output a 4-20mA analog signal corresponding to a change in stimuli. The pH sensor works by using the principle that having a high hydrogen ion concentration increases the potential to conduct electric current. This potential, E, is defined by the Nernst Equation given in Eq. 1 where E_o and H^+ are the standard potential of the hydrogen electrode and the hydrogen ion concentration in the sample respectively.

$$E = E_o + 0.0059\ ln[H^+]\ V \tag{1}$$

The voltage difference is measured by the use of a measurement electrode and a reference electrode and is transmitted by the pH sensor in the form of a 4-20mA signal. A 250-ohm resistor is placed in this 4-20mA current loop and the voltage across this resistor is fed into the analog pins of the AtMega 2560 microcontroller. This voltage is mapped out as integers from 0 to 1023 by the Arduino's 10-bit Analog to Digital Converter (ADC). This value is then correlated to the pH scale in the microcontroller after adjusting its calibration constant.

The conductivity sensors apply an alternating current (I) at an optimal frequency to two active electrodes and measures the potential (V). The current and potential are used to calculate the conductance (I/V). The conductance, along with the cell constant is then used to calculate the conductivity of the solution under observation as shown in Eq. 4. The cell constant depends on the physical dimensions of the conductivity sensor like the difference between the electrode plates, d and the surface area of the electrode plates a as shown in Eq. 3.

$$Conductance\ (S),\ G = I/V \tag{2}$$

$$Cell\ Constant\ (cm^{-1}),\ K = d/a \tag{3}$$

$$Conductivity\ (S/cm^{-1}),\ \kappa = G.K \tag{4}$$

The communication is similar to that of the pH sensor. The conductivity sensor transmits a 4-20mA current signal which is then converted to a voltage signal to be read by the ADC of the AtMega 2560 microcontroller. The analog signals from both the conductivity and the pH sensors are passed through a signal conditioning circuitry before A/D conversion.

Magnetic float sensors consist of a magnetic reed switch in the stem and a magnetic in its bulb. When water fills up in the tank, the bulb structure with the magnet moves up to the stem, switching ON the reed switch. These sensors send a simple digital ON/OFF signal based on the position of the magnetic bulb. Optical isolation has been provided to the digital inputs to prevent ground looping. MCT2E, a phototransistor based optocoupler has been used to separate the grounds of the data lines and the power lines.

After all the sensor readings are acquired and processed, a data string with all the measurements along with the corresponding timestamp is created in the microcontroller and stored in *.csv* format locally in the micro SD card.

2) Microcontroller Node to Web Dashboard: The data string is sent to the IoT gateway through a serial port for converting the data semantics into a JavaScript Object Notation (JSON) format which is supported by the Kaa platform. Messaged Queued Telemetry Transport (MQTT) protocol is used to transfer the data over the internet to the IoT platform. MQTT is chosen because it is a lightweight messaging protocol for IoT applications that uses a publish and subscribe model. Compared to HTTP, the MQTT protocol is designed for low bandwidth networks and can send high volumes of sensor messages to cloud-based platforms. The IoT gateway is designated as an MQTT broker which publishes data to a topic. Any client, like the IoT Platform in this case which subscribes to that particular topic can stream the data as and when it is published.

C. Operation

The data fetching, processing and storage algorithm of the EMCS runs in a continuous loop as shown in Fig. 3. On startup, the microcontroller initializes the libraries required to interface the RTC and SD Card Module. The microcontroller then enters the loop where the data from all the sensors is measured, processed and stored. A modular architecture is adopted for the software too. The code is broken into several modules or functions, each performing an independent task. This allows for better flexibility and easy debugging. For example, the *check_level()* function checks the status indicator from all the magnetic level sensors. Flags are then set for all the tank level statuses.

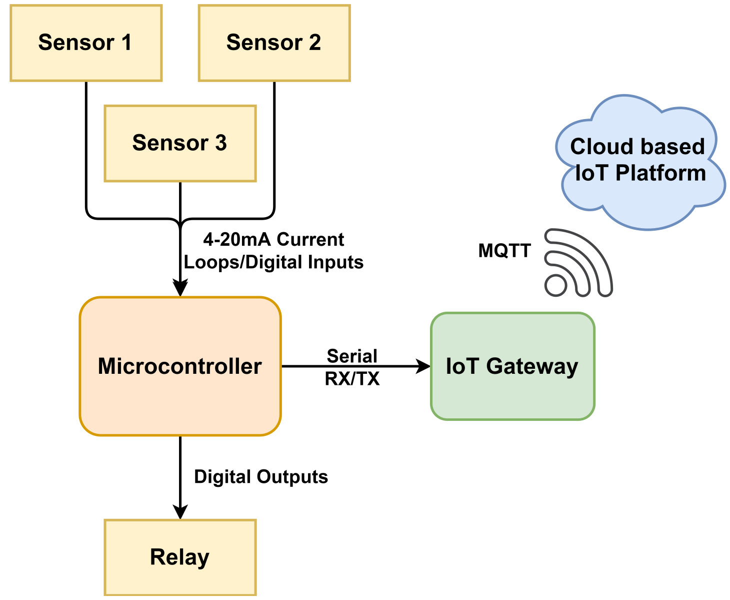

Fig. 2. Sensor Data Flow in the EMCS

Analog sensors like conductivity, pH and pressure sensors are then read by the ADCs of the AtMega 2560 microcontroller. Ten readings are taken by the microcontroller from each analog pin at a gap of 30ms. These readings are stored in an integer array and then passed through a bubble sort algorithm to arrange them in ascending order. The readings at the extreme ends are neglected and the average of the remaining readings is calculated for mapping to the range of the sensor. This is done to avoid taking into account any potential extraneous data acquired by the sensor. Calibration factors, offsets and sensor biases are taken into account before the final mapping.

Some base sensors can be used in combination to derive several other parameters. For example, the value of conductivity can be used to derive other parameters like Total Dissolved Solids, (TDS) as shown in Eq 5. Since conductivity is a highly temperature-dependent parameter, a water quality parameter called Specific Conductance is used. Specific conductance, EC_{25}, is the conductivity of the sample at room temperature as shown in Eq. 6. The room temperature, t_{room}, and the conductivity at room temperature, EC_{room} is used to calculate specific conductance.

$$Conductivity = \frac{TDS}{Conversion\ Factor} \tag{5}$$

$$EC_{25} = \frac{EC_{room}}{1 + \alpha(t_{room} - 25)} \tag{6}$$

The conversion factor to calculate TDS depends on the type of minerals and salts dissolved in the water sample being monitored.

The microcontroller now opens the micro-SD card and checks if there is a file for the current date. If there isn't, a new file is created. After each cycle of reading the sensor data, the

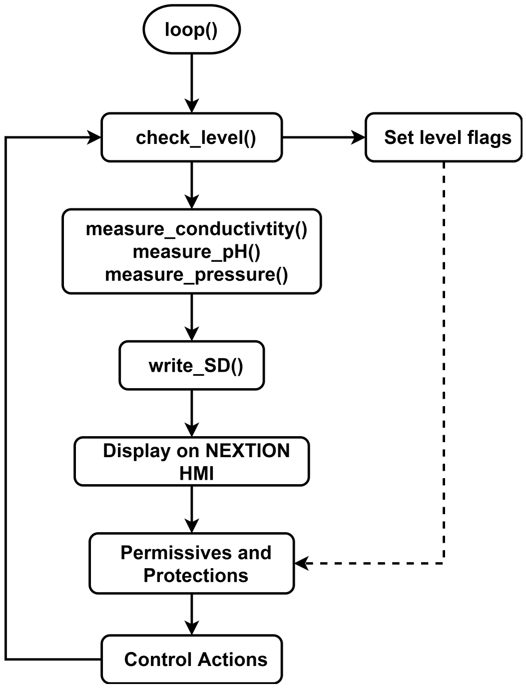

Fig. 3. Operating loop for sensor data acquisition

values are appended to a data string along with the timestamp. This data string is written on a file with that day's data. A new file is created every day to identify the operation of the plant. This data string is then sent to the IoT gateway using the serial port. The string is split and its components are then arranged in a JSON dump to be sent to the web dashboard. Every sensor data is assigned to a unique identifier.

The IoT gateway is registered as an endpoint in a custom-designed Kaa application. By using the unique endpoint token and the application key, a connection can be established between the IoT gateway and the Kaa platform. Once the connection is established, the serial data flushed into the gateway will be published by the gateway (acting as an MQTT broker). All the clients subscribing to that particular topic will subscribe to the data.

IV. Implementation

With the basic architecture for an IoT based remote monitoring and control system set up, we started interfacing locally available low-cost conductivity, pH and level sensors to the microcontroller. A custom UI was designed using the Nextion Editor with a central menu page and individual pages for each of the monitored parameters as shown in Fig. 4.

Before sending the data to the cloud, the IoT gateway is registered as an endpoint in the custom Kaa application. Once the connection is established and verified, the constant stream of sensor data is acquired, normalised, converted to JSON format and transmitted to the cloud. Widgets like time-series graphs, gauges, and real time tables can be added to the dashboard for every JSON identifier. The incoming data stream is automatically sorted by the Kaa engine and each JSON

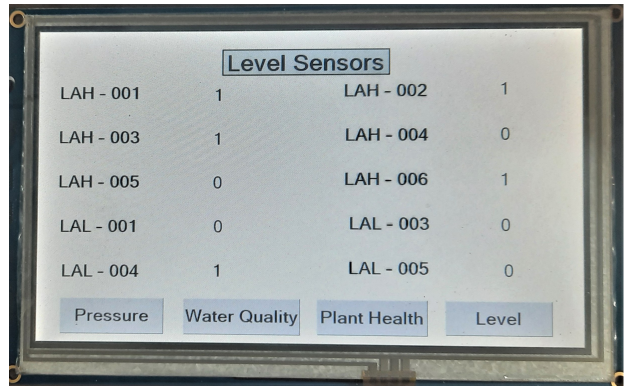

Fig. 4. Touchsceen HMI for local display of plant parameters

object can be assigned to the widget for visualisation. Fig. 5 shows one such widget for monitoring the pH of the water line monitored. Kaa also supports the creation of custom widgets and controls enabling the creation of complex dashboards.

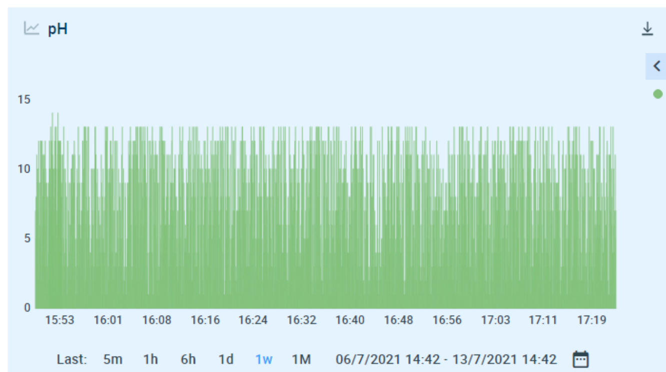

Fig. 5. pH Monitoring on Kaa dashboard

Widgets like tables, line graphs, bar charts, gauges and digital meters are used to visualise the streamed data and create a dashboard for the end user as shown in 6. Keycloak, an open source access management software is used to grant admin and tenant access to specific users.

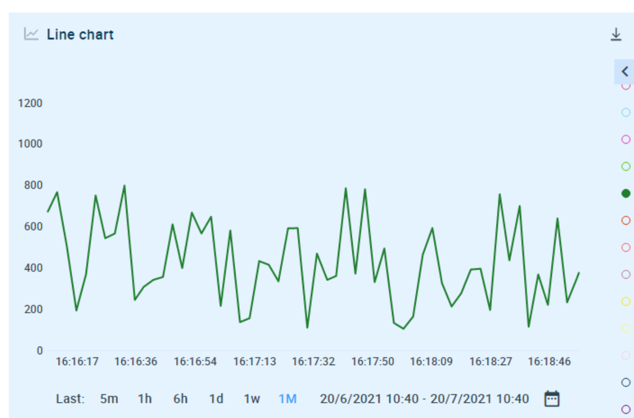

Fig. 6. Line Chart on the Kaa dashboard

V. Conclusion

The basic architecture for IoT based remote monitoring and control system allows for a wide variety of analog and digital sensors to be incorporated into it. Once the measured values are normalised to the standard range of the sensors, they will be sent to the IoT gateway where it is arranged in a JSON format. The advantage of using the Kaa platform is that it offers a seamless integration to OpenDistro's stack allowing us to set various monitors and alerts. Alerts can be set for Water quality parameters like conductivity and pH, and when it crosses a pre-defined threshold, an alert will be triggered. Tank levels can also be similarly continuously monitored and pump actions can be similarly controlled. Kaa has a provision for multitenancy which is useful for scaling up. A single IoT gateway can be integrated with multiple endpoints saving up on infrastructure costs.

Real-time analytics like optimising pump operation cycles and long term scheduling can be performed by regularly collecting plant operation data. Intelligent algorithms and rule engines to detect leaks and bursts and pump breakdowns exist in literature and can be actualised. Machine Learning algorithms to predict membrane changes and potential malfunctions can be also be implemented. A rule engine to operate the Water Purification and Recycling plant based on the tank levels and pressure in the lines can be programmed and implemented. Protections to prevent water wastage and plant component safety can be enforced.

The developed Electronic Monitoring and Control System can provide local as well as cloud-based monitoring and control. It also makes it possible to store huge amounts of plant data locally as well as in the cloud for further analytics. The developed EMCS is simple to operate and and it's utility can be bolstered by the addition of different sensors for measuring other contaminants. As both the hardware and software are modular, by making small relevant changes, the necessary sensors can be easily incorporated. This type of low-cost solution can replace more expensive monitoring and control systems for operating smaller low-cost Water Purification and Recycling plants.

Acknowledgment

This project has received funding from the Indian Department of Biotechnology and the European Union's Horizon 2020 research and innovation programme under grant agreement No 820906. The authors would also like to thank CSIR - Central Electronics Engineering Research Institute (CEERI) for providing the necessary infrastructure to develop this project.

References

[1] Jan, Farmanullah, Nasro Min-Allah, and Dilek Düştegör. "IoT Based Smart Water Quality Monitoring: Recent Techniques, Trends and Challenges for Domestic Applications." Water 13.13 (2021): 1729.

[2] Serpanos, Dimitrios. "The cyber-physical systems revolution." Computer 51.3 (2018): 70-73.

[3] Adu-Manu, Kofi Sarpong, et al. "Water quality monitoring using wireless sensor networks: Current trends and future research directions." ACM Transactions on Sensor Networks (TOSN) 13.1 (2017): 1-41.

[4] Cloete, Niel Andre, Reza Malekian, and Lakshmi Nair. "Design of smart sensors for real-time water quality monitoring." IEEE access 4 (2016): 3975-3990.

[5] Kodali, Ravi Kishore, Sasweth C. Rajanarayanan, and Sindhuja Yadavilli. "IoT Monitoring Setup for Waste Water Treatment." 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129). IEEE, 2019.

[6] Aggarwal, Sachin, Rahul Gulati, and Bharat Bhushan. "Monitoring of input and output water quality in treatment of urban waste water using IoT and artificial neural network." 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). Vol. 1. IEEE, 2019.

[7] Hayder, Gasim, Isman Kurniawan, and Hauwa Mohammed Mustafa. "Implementation of machine learning methods for monitoring and predicting water quality parameters." Biointerface Res. Appl. Chem 11 (2020): 9285-9295.

[8] Nadiri, Ata Allah, et al. "Prediction of effluent quality parameters of a wastewater treatment plant using a supervised committee fuzzy logic model." Journal of cleaner production 180 (2018): 539-549.

[9] Zidan, Nael, Mohammed Maree, and Subhi Samhan. "An IoT based monitoring and controlling system for water chlorination treatment." Proceedings of the 2nd International Conference on Future Networks and Distributed Systems. 2018.

[10] Budiarti, Rizqi Putri Nourma, et al. "Development of IoT for automated water quality monitoring system." 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE). IEEE, 2019.

[11] Gopavanitha, K., and Sholipuram Nagaraju. "A low cost system for real time water quality monitoring and controlling using IoT." 2017 International conference on energy, communication, data analytics and soft computing (ICECDS). IEEE, 2017.

[12] Shevale, Rupalir, et al. "IOT Based Real time water Monitoring System for Smart City." International Journal of Innovative Science and Research Technology, ISSN 2456-2165.

[13] Pappu, Soundarya, et al. "Intelligent IoT based water quality monitoring system." International Journal of Applied Engineering Research 12.16 (2017): 5447-5454.

[14] Chen, Yiheng, and Dawei Han. "Water quality monitoring in smart city: A pilot project." Automation in Construction 89 (2018): 307-316.

[15] Martínez, Ramón, et al. "On the use of an IoT integrated system for water quality monitoring and management in wastewater treatment plants." Water 12.4 (2020): 1096.

[16] Wang, Charlie. "HTTP vs. MQTT: A tale of two IoT protocols." Julkaistu 26 (2018): 2018.