Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 07 (July -2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i07.pp01 - 07*

Scalable and Robust IoT-Based Liquid Level Monitoring/Control System for Home and Industrial Automation

Prasanth P. P.[1] | Athul Thomas[1] | James Kurian[1]

[1]Department of Electronics, Cochin University of Science and Technology
Cochin-682 022,

To Cite this Article
Athul Thomas1, James Kurian **, Scalable and Robust IoT-Based Liquid Level Monitoring/Control System for Home and Industrial Automation "** *Journal of Science and Technology, Vol. 08, Issue 07,-July 2023, pp01-07*

Article Info
Received: 22-06-2023 Revised: 25-06-2023 Accepted: 02-07-2023 Published: 08-07-2023

ABSTRACT

In this study, a dependable liquid level monitoring and control system for automated home or industrial settings is presented. The system uses state of the art Internet of Things (IoT) technology to accomplish its goals. The suggested system, which consists of a minimum of four interconnected functional units, utilises Wi-Fi connectivity to establish communication between the parts. These nodes can efficiently share data by using the simple MQTT protocol, which enables scalability by allowing the installation of additional nodes to manage various tank and reservoir levels. In the suggested approach, the Sensor node simultaneously subscribes to several topics pertaining to user settings and measurement frequency while publishing measurements to the server. By subscribing to these topics, the Actuator Control node may control pumps or valves appropriately and communicate their current states. The User Interface node publishes changes to user configuration and subscribes to all topics from other nodes. All firmware codes were created using the C++ libraries and programming structures in the Arduino development environment. An Android application that has been specially created can serve as the system's graphical user interface, allowing users to interact with it via a variety of portable devices.
KEYWORDS: *Internet of Things, MQTT, Arduino, Home automation*

INTRODUCTION

Liquid level management is a common requirement in many industrial environments. For automatic liquid level monitoring and control, it is desirable to have a reliable system which adds less complexity to the existing factory automation set up. In conventional level monitoring/controlling systems [1-3], their installation and maintenance are tedious and laborious. It is mainly due to the fact that the instrumentation demands a tangled cabling for the interconnection of the system. This includes the sensing, processing and control units, that relies in different sections of the factory floor. The presented work focuses on a feasibility study of implementing state of the art IoT (Internet of Things) technology for the remote monitoring and controlling [4-7] of liquid level in an industrial or home automation scenario.

The next section discusses the main objectives of the system that were considered during the designing process. It follows an overview of different units of the level controller and discusses the hardware of each node. A detailed explanation of the communication protocol and the software and app development were provided in the software section. The final section concludes the work and unveils the future scopes and modifications.

Figure 1 Shows the block diagram representation of different nodes of the proposed liquid level monitoring and control system

I. OBJECTIVES

The main objective of this study is to develop and implement a liquid level monitoring and control system, and to extend its features using the capabilities of IoT technology. Recent advancements in wireless technology and Chip manufacturing offers Wi-Fi enabled low power microcontroller units. These units are handy to control peripherals like sensors; relays etc. and allow them to communicate over a wireless network. Some of the target requirements of the system architecture includes:

Distributed Wireless Computing Nodes: Here the elementary units of the system, such as sensors, actuators, user interface etc. were implemented in fully independent, self-contained hardware modules. This feature improves the modularity of the system so that any number of nodes can be added (as a part of system upgradation) or removed (during maintenance) from the architecture without disturbing its overall functionality.

Client Server Architecture: Here, as the name indicates, the whole system is divided in to a group of clients and a main server. Clients are intended to do a specific dedicated operation like sensing, actuator controlling, etc., where there is less requirement for processing power, memory and I/O operations. Processes such as network management and plant operations log, can be moved to the server node with better computational resources. This approach also helps to improve the cost effectiveness of the system by using low cost processors for the client modules.

Scalability: As discussed before, the architecture of the system should be dynamic so that any elementary module can be inserted or removed in the system. This feature will free the developer from changing the entire firmware or configuration each time, when a new node is added.

II. SYSTEM HARDWARE

Proposed hardware model for the liquid level monitoring and control system has been divided into four functional nodes, as depicted in Figure 1.

i Level Sensing
ii Actuator Control
iii User Interface
iv Server

Raspberry pi 2 has been used as the server to control the network. The remaining nodes were controlled by ESP8266-12F Wi-Fi module. Low-power, highly-integrated ESP8266-12F modules enables the implementation of wireless network. Client nodes were programmed in the station mode (STA). The ultra-compact power supply module HLK-PM01 facilitates 5v dc output from the main supply and ensures proper isolation. Following subsections explains each of the nodes in detail.

Level Sensing

The Level Sensing node includes an economic ultrasonic sensor HC-SR04 along with an ESP-12F Wi-Fi module programmed in the station mode (STA). Compared to traditional level sensors like electrodes or float switches, non-contact ultrasonic sensors offer high degree of accuracy and reliability. The HC-SR04 module comprise ultrasonic sound transmitter, detector and control circuit. The transmitter sends out sonic bursts which will travel at the speed of sound and the echo from liquid is received by detector. By estimating the time of flight using a microcontroller, the level of liquid can be determined. The sensor module can provide 2cm to 400cm [8] of measurement range with a resolution of up to 3mm. The hardware is mounted on top of the container with ultrasonic module facing towards the liquid.

Besides the Wi-Fi functionalities ESP8266 integrates an enhanced version of Tensilica L106 microcontroller. The controller offers enough I/O modules and A-D converters to interface sensors or other peripherals. The Trigger, Echo pins from the Ultrasonic module HC-SR04 is connected to the programmable I/O pins of the ESP8266 module. The L106 controller estimates distance towards obstacle by analysing the delay between trigger and echo pulses. Measured levels will be scaled and calibrated according to the container dimensions. Depending on the user configuration, this value may be processed further before getting transmitted to the server. Figure 2 shows the Level Sensing node.

Figure 2. Shows the photograph of Level Sensing node with ultrasonic sensor and other programming and interfacing circuits

Actuator Control

The Actuator Control node integrates an appropriate power driving system for the pump/valve control. ESP8266 Wi-Fi module receives commands from the server through Wi-Fi and act accordingly by driving the relevant actuator. Figure 3 shows the Actuator Control node.

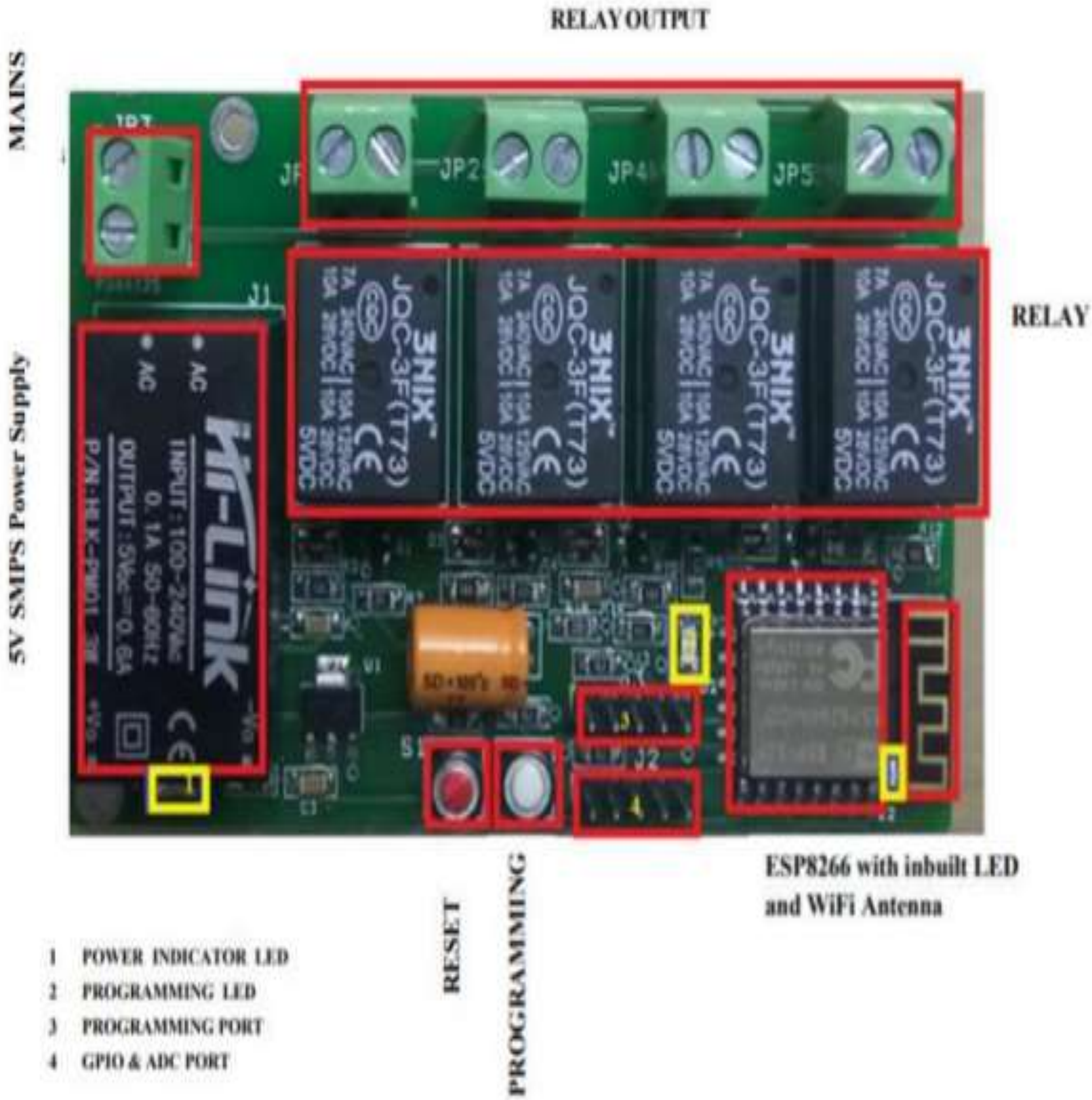

Figure 3. Shows the photograph of Actuator Control node with four relays and connectors with driving circuits for valves/pumps

As shown in the figure, the Actuator Control node has four 5V relays with a rating of 10A. These four channels can be either configured as independent mono switches or as a pair of dual switches. This feature enables the node to drive different models of valves and motors.

User Interface

In an industrial system, there should be some provision for human machine interaction. The level control system has several parameters such as maximum and minimum level, flow rate, manual on/off, current liquid level, etc., for monitoring and controlling. This can be made possible through the following ways.

User can interact with the level controller system in the following ways. The server provides a terminal where the control parameters can be adjusted and current system status can be observed. A custom developed GUI app can communicate with the server and display the information on a portable mobile device. A dedicated hardware with OLED and push buttons with a microcontroller will serve as a User

Interface node that provides an aid for user interaction for controlling or monitoring the system parameters.

Server Unit

The Server Unit (MQTT broker), handles and relays communications between all the other units. This work uses Raspberry Pi with a Linux operating system as the MQTT broker [9].

The Level Sensing Unit is linked to both the User Interface Unit and to a process control routine running in the Actuator Control Unit via local Wi-Fi network. The measured liquid level is wirelessly transmitted to the process control to adjust the control actuators. The level data is also shared with the User Interface unit for monitoring the instantaneous liquid level. The communication between the sensor unit, control unit and the user interface unit has been implemented using Message Queuing Telemetry Transport (MQTT) protocol, that is explained in the software section below.

III. SYSTEM SOFTWARE

The most important part of the software development was to find a suitable IoT protocol on which the entire system communications can be relied. In the proposed system the Message Queuing Telemetry Transport (MQTT) protocol has been used.

All the firmware codes for the client nodes were developed in the Arduino development environment and there by uses the C++ libraries and programming structures. An android application has also been developed to be used as a convenient mobile interface through which the user can interact with the system from various mobile devices.

MQTT Protocol

Message Queue Telemetry Transport is a lightweight connectivity protocol based on the TCP/IP stack which uses the publish/subscribe method for transportation of data [10-12]. It is open-ended and supports a high level of scaling, which makes it an ideal platform for development of Internet of Things (IoT) solutions. It is a message-based protocol that uses publisher-subscriber pattern. The key component in MQTT is the MQTT broker. The main task of MQTT broker is dispatching messages to the clients (“subscribers”). Broker receives messages from publisher and dispatches these messages to the subscribers. While it dispatches messages, the MQTT broker uses the topic to filter the clients that will receive the message. The topic is a string and it is possible to combine the topics creating topic levels. Figure 4 shows an overview of the MQTT protocol.

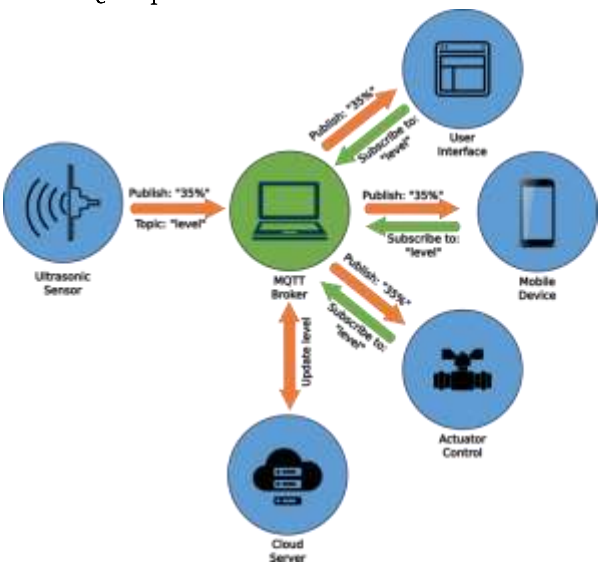

Figure 4. Shows the working of the system with MQTT protocol for updating liquid level

Client Firmware

The open source Arduino Integrated Development Environment (IDE) has been used for the firmware development of all client boards. Several libraries were used for the programming and driver development. Some of them have been modified to meet the requirements discussed in the objectives.

Android App Development

For the android app development MIT App Inventor has been utilized [13-16]. It is an open source web application, developed by MIT Media Lab and MIT Computer Science and Artificial Intelligence Lab, to create software for the Android operating system. It uses an easy graphical interface that allows users to drag and drop visual objects to create an application that can run on Android devices. Figure 5 depicts a screenshot of the Android app.

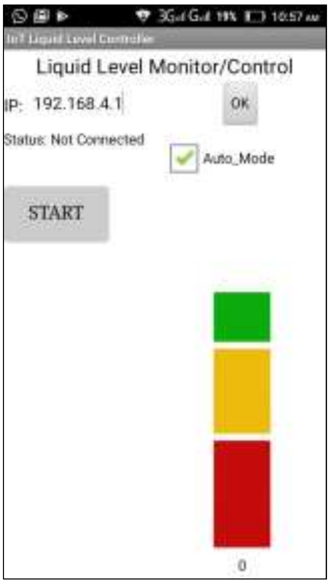

Figure 5 Shows the screenshot of the level controller application with the current level and pump status

Server Implementation

The Raspberry Pi board with a Linux OS acts as the server (MQTT Broker) for the entire system. The MQTT protocol has been implemented in the server with Eclipse Mosquitto. Eclipse Mosquitto provides a lightweight server implementation of the MQTT protocol that is suitable for the low power embedded computers. The current implementation has the capability to accept the connections from up to 1000 clients. It also has an option to enable a bridge that allows it to connect to other MQTT servers, including other Mosquitto instances. This allows networks of MQTT servers to be constructed, passing MQTT messages from any location in the network to any other, depending on the configuration of the bridge.

Figure 6. Shows the working of the system with MQTT protocol for updating liquid level

Cloud Connection

Communicating sensor readings, switch states etc with the internet is the key point in any IoT enabled system. The proposed level controller uses Adafruit Cloud as the remote server (Figure 6). The cloud facilitates a secured environment where the received sensor data can be analysed and based on that the control valves can be adjusted [9]. Server running in Raspberry Pi will talk with Adafruit cloud over the internet. Adafruit dashboard has feeds corresponding to “sensor level”, “switch state” etc. The local server can publish its readings or subscribe to switch states through these feeds. Any device which has access to the Adafruit IoT dashboard can view or modify the system once authenticated with the API key.

III. CONCLUSION

Implementation of an IoT based liquid level monitoring/controlling system is discussed. The MQTT protocol has been successfully integrated in the firmware design that enables an easy interfacing and installation of hardware nodes without any major modification of the existing factory management setup. The same implementation can also be used in home automation with several added features such as optional manual override, water consumption rate.

REFERENCES

[1] Beza N. Getu, Nasser A. Hamad, Hussain A. Attia, "Remote Controlling of an Agricultural Pump System Based on the Dual Tone Multi-Frequency (DTMF) Technique", Journal of Engineering Scienceand Technology (JESTECH), vol. 10, no. 10, pp. 1261-1274, 2015.

[2] C.N. Anyanwu, C.C. Mbajiorgu, E.C. Anoliefo, “Design And Implementation of a Water Level Controller”, Nigerian Journal of Technology, Vol 31, March 2012, pp. 89-92.

[3] Beza Negash Getu, Hussain A. Attia, "Automatic water level sensor and controller system," 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, 2016, pp. 1-4.

[4] Shopan Dey, Ayon Roy, Sandip Das, "Home automation using Internet of Thing", IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, 2016, pp. 1-6.

[5] Pavithra, D., and Ranjith Balakrishnan. "IoT based monitoring and control system for home automation." 2015 global conference on communication technologies (GCCT). IEEE, 2015.

[6] Kodali, Ravi Kishore, et al. "IoT based smart security and home automation system." 2016 international conference on computing, communication and automation (ICCCA). IEEE, 2016.

[7] Abdulraheem, Ahmad Sinali, et al. "Home automation system based on IoT." Technology Reports of Kansai University 62.5 (2020): 2453.

[8] HC-SR04 datasheet, October 2018, [online] Available: www.mouser.com/ds/2/813/ HCSR04-1022824.pdf.

[9] Mosquitto installation on Raspberry Pi, Aug.2018, [online] Available: www.instructables.com/id/ Installing-MQTT-BrokerMosquitto-on-aspberry-Pi/.

[10] MQTT Protocol, Aug.2018, [online] Available: en.wikipedia.org/wiki/MQTT

[11] Gupta, Vatsal, Sonam Khera, and Neelam Turk. "MQTT protocol employing IOT based home safety system with ABE encryption." Multimedia Tools and Applications 80.2 (2021): 2931-2949.

[12] Atmoko, R. A., R. Riantini, and M. K. Hasin. "IoT real time data acquisition using MQTT protocol." Journal of Physics: Conference Series. Vol. 853. No. 1. IOP Publishing, 2017.

[13] Oliveira, Guilherme MB, et al. "Comparison between MQTT and WebSocket protocols for Iot applications using ESP8266." 2018 Workshop on Metrology for Industry 4.0 and IoT. IEEE, 2018.

[14] Panwar, Ayush, et al. "Eyrie smart home automation using Internet of Things." 2017 computing conference. IEEE, 2017.

[15] MIT Center for Mobile Learning. MIT App Inventor Public Open Source, https://github.com/mit-cml/appinventor-sources, accessed Sep. 2, 2014.

[16] P. Kishore, T. Veeramanikandasamy, K. Sambath, S. Veerakumar, “Internet of Things based Low-Cost Real-Time Home Automation and Smart Security System”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, pp. 505-509, April 2017.