Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System

Conference Paper · September 2020

DOI: 10.1109/IEMTRONICS51293.2020.9216412

2 authors:

Atefeh Zare
Memorial University of Newfoundland

M. Tariq Iqbal
Memorial University of Newfoundland

Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System

Atefeh Zare
Department of Electrical and Computer Engineering,
Faculty of Engineering and Applied Science
Memorial University of Newfoundland (MUN)
St. John's, NL A1B 3X5, Canada
* Correspondence: Email: azare@mun.ca

M. Tariq Iqbal
Department of Electrical and Computer Engineering,
Faculty of Engineering and Applied Science
Memorial University of Newfoundland (MUN)
St. John's, NL A1B 3X5, Canada
tariq@mun.ca

***Abstract*—Growing energy cost and demand has motivated many organizations to achieve smart ways to monitor, control, and save energy. Smart automation can reduce costs while still satisfying energy demand. The residential, commercial, and industrial sectors can utilize the technologies of the Internet of Things (IoT) to manage energy consumption better. This paper presents a low-cost, open-source, and reliable Supervisory Control and Data Acquisition (SCADA) system for home monitoring and control system. The presented SCADA system consists of analog sensors, ESP32, Node-RED, and Message Queuing Telemetry Transport (MQTT) through local Wi-Fi to remotely access and control appliances. This system helps the users to monitor various conditions in the home, such as temperature, humidity, pressure, and light intensity. Thus, users can remotely monitor various devices such as lights, fans, heating/cooling systems, make decisions based on the feedback of sensors.**

***Index Terms*—ESP32, Internet of Things (IoT), Wi-Fi network, MQTT, Node-RED, Sensors**

I. Introduction And Literature Review

NOWADAYS, the fraction of automation systems in the residential sector is rising rapidly because of numerous life improvements such as comfort, convenience, centralized control of appliances, cost reduction, energy-saving, security, and safety. As a home automation system enhances the life quality for users, especially for the elderly, differently abled persons, and people who want to monitor and manage their home devices' operation, it is quietly important to have a proper control system. SCADA technologies implement a unique platform that senses and monitors remote devices, acquires data from them, and sends limited control instructions. Besides that, this system allows users to discover the status of devices and their residence conditions remotely.

Home automation systems using SCADA consists of four major parts. The first part is the sensing devices placed at several locations throughout the home to measure the desired variables. The second part is Remote Terminal Units (RTUs) for acquiring remote data from sensors. The third one is Master Terminal Units (MTUs) to process the received data and deal with Human Machine Interfaces (HMI) [1]. The last part of SCADA is the communication channel to connect the RTUs to the MTUs [2 ,3]

SCADA technology has been developed over the past years to remotely monitor and control processes. In this work, an open-source and low-cost SCADA system based on the Internet of Things is introduced. This SCADA system utilizes reliable and commonly available components to fulfill the four main functions of a SCADA system: Data collection, network data communication, data presentation, and remote monitoring and supervisory control [4].

Several researchers around the world have designed SCADA systems based on IoT architecture in the past. Lekic et al. [5] performed an IoT-based SCADA system with the Raspberry Pi3. Temperature and humidity sensors acquired the desired data, and the IBM Bluemix cloud platform was utilized to receive, visualize, and manage the collected sensor data via the web while using the Node-Red and Web Socket protocols for data exchange and communication between the cloud platform and sensors connected to the Raspberry Pi.

In [6], Rai et al., Using IoT, implemented a system to provide the end-users with a cost-effective and portable intelligent monitoring system and enable devices remotely. The proposed system required a low-resolution video camera module interfaced with the Arducam ESP32 UNO board. The resulting video was streamed using ESP32 integrated Wi-Fi to display on the 1.8-inches SPI TFT Adafruit display module.

M. Al-Kuwari et al. [7] demonstrated sensing and monitoring platform on smart and IoT-based home automation system, where the authors presented a basic concept of how home automation can be deployed using IoT.

Pravalika et al. [8] implemented a home automation system through the Wi-Fi module, Massachusetts Institute of Technology (MIT) app, and a web page server using ESP32 to monitor home devices.

In [9], the MQTT-based home automation system using ESP8266 Node MCU was established for remote monitoring and controlling through a standard gateway. This system was designed utilizing the GSM (Global System for Mobile Communications) network.

In [10], Kodali demonstrated IoT-based smart environment monitoring using Arduino and C embedded programming devices, which were remotely controlled with the Internet. Temperature, humidity, light level, vibration, and air quality sensors connected to the controller for measurement and

ESP8266 Node MCU for the Wi-Fi network.

Pujari et al. [11] demonstrated a smart home system that uses Wi-Fi to connect to the Internet, remotely monitor and control the appliances, and surveillance. Sensors were connected to ESP32 to collect data. All acquired data was uploaded to Firebase via the ESP32's built-in Wi-Fi, making it possible to control the home environment using the developed applications.

In [12], the IoT-based SCADA system proposed which the Raspberry Pi3 and Intel Edison board used for acquiring sensor inputs. Collected sensor data are transmitted to Amazon Web Services (AWS) through MQTT brokers in Node-Red. On the AWS IoT platform, several monitoring and control schemes inducted using Amazon's Voice Service named Alexa.

In this work, a low-cost and open-source SCADA system is developed that uses Wi-Fi, the MQTT protocol, and Node-Red. The proposed system can provide detailed measurements of temperature, pressure, humidity, and light density. Besides that, the states of the devices at home are being traced. Therefore, users can understand the status of their home devices such as lights, fans, air conditioners, or heating/cooling systems then remotely control them.

II. System Design

The proposed SCADA system block diagram is presented in fig.1. It is implemented using ESP32 as the sensor gateway and RTU and Raspberry Pi2 as a local server. SparkFun Atmospheric Breakout Sensor-BME280 sensor and an LDR are connected to ESP32 to acquire the desired data and the dashboard to receive, visualize and manage the acquired sensors data over the web while using Node-Red and MQTT protocol for data exchange and communication between the MQTT clients such as end-users and the ESP32 connected sensors while the acquired data are being stored in SQLite as a web server. Each component is described in detail as following:

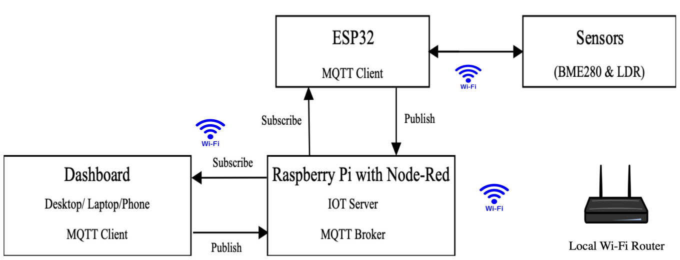

Fig. 1. The proposed SCADA system block diagram.

A. MQTT Communication Protocol

As shown in Fig. 2, MQTT is a publish/subscribe protocol. It is a lightweight and straightforward messaging tool designed for unreliable networks and constrained devices and low bandwidth. It is a suitable solution for this design since it provides simple communication between the server (MQTT broker) and clients (ESP32 microcontroller and computers and mobile devices) [9-10, 13]. Clients can subscribe to the topics or publish the data to topics of any kind of data. The broker then distributes the data to any client that has subscribed to that topic. Eclipse Mosquitto software is being run as a broker on the local server (Raspberry Pi2).

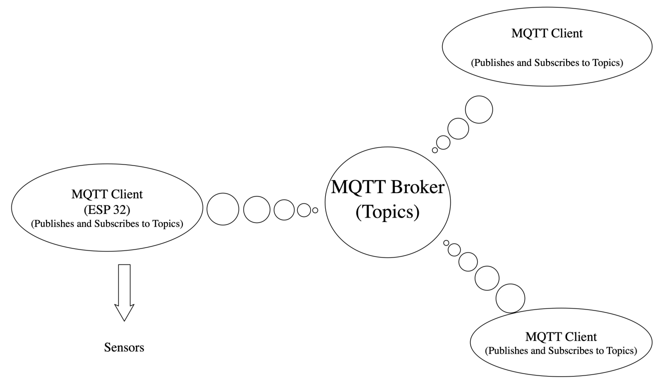

Fig. 2. MQTT Architecture.

In this project, the ESP32 microcontroller publishes the sensors data with specific topics to the MQTT Broker (Mosquitto Broker), while personal computers and mobile devices subscribe to the topics to visualize the published data on the server. MQTT protocol makes the supervisory control possible in this project.

B. Raspberry Pi

The Raspberry Pi2 model B is used as the local server at a local network, including MQTT broker, Node-Red, SQLite.

C. ESP32 Thing

The ESP32 Thing is developed by SparkFun Electronics. It is one of the most low-cost, low-power, and small microcontrollers included Wi-Fi and Bluetooth modules. Fig. 3 shows a picture of the SparkFun ESP32 board.

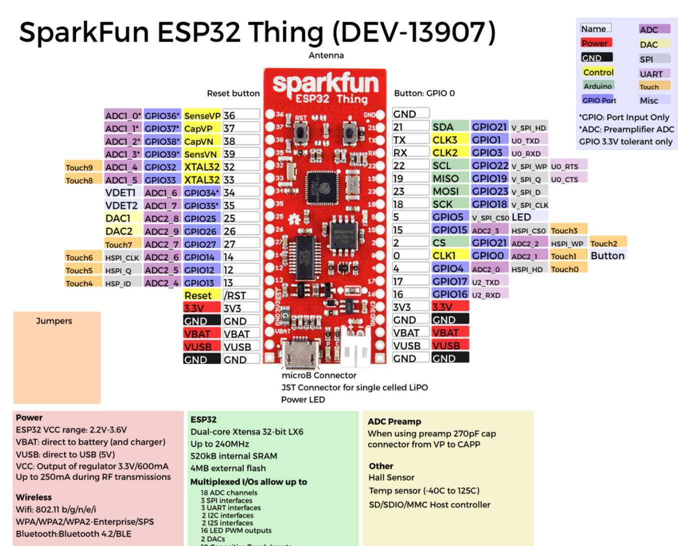

Fig.3. Pin description of ESP32.

The board operating signal voltage range is 2.2 V to 3.6 V, and it can be supplied with either a 5 V USB power supply or with a battery. The microcontroller is programmed with the Arduino software integrated development environment (IDE). The Arduino IDE allows experts to write the desired programs and upload them to the board via a USB cable. In this work, the ESP32 microcontroller is programmed as an MQTT Client using the Arduino IDE software. The MQTT Client Library called PubSubClient. The program is such that the board collects the measured real-time values of desired variables, displays the values on the Arduino IDE Serial Monitor, and continuously publishes real-time data to the MQTT Broker.

D. SparkFun Atmospheric Sensor Breakout BME280

The SparkFun BME280 Atmospheric Sensor Breakout is used to measure pressure, humidity, and temperature readings, all with a tiny breakout. It is shown in Fig. 4. The BME280 Breakout is designed to be deployed in indoor/outdoor navigation, home automation, etc. The on-board BME280 sensor measures atmospheric pressure, humidity, and temperature. The breakout presents a 5V tolerant I2C interface (with a pull-up resistor to 3.3V to be cooperative with ESP32).

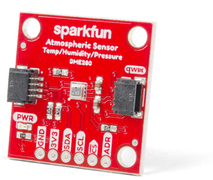

Fig.4. SparkFun Atmospheric Sensor Breakout BME280.

E. Light Dependent Resistor (LDR)

The LDR is a variable resistor. Fig. 5 presents LDR and its step-down resistor connection. The resistance of the LDR inversely proportionates to the light intensity. It shows maximum resistance in the absence of light and minimum resistance in the presence of light. Considering this characteristic, in this project, during the nighttime, LDR turns the LED ON and reversely makes the LED OFF during the daytime.

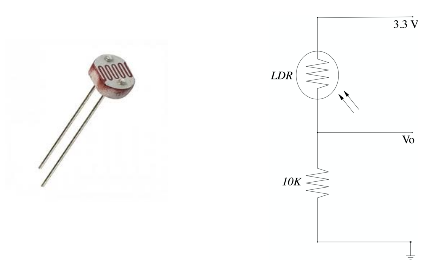

Fig.5. LDR and its step-down resistor connection.

The relationship between the resistance R_L and light intensity Lux for a typical LDR is [14]

$$R_L = \frac{500}{Lux} \tag{1}$$

With the LDR connected to 3.3V through a 10K resistor, the output voltage of the LDR is

$$V_O = \frac{10*R_L}{(R_L+10)} \tag{2}$$

From equation (1) and (2), light intensity obtained as follows:

$$Lux = \frac{500}{V_O} - 50 \tag{3}$$

The obtained Lux value is being used for controlling the LED.

F. Node-Red

Node-RED is an open-source programming tool used for wiring together hardware devices, APIs, and online services smartly [5]. It can be installed on a Linux-based platform, and it provides a browser-based editor that makes it easy to wire together flows using various nodes in the palette that can be deployed to its runtime.

G. Database and dashboard

An SQLite database using the litedb node on the Node-RED platform is installed to create database tables and store data. As a very lightweight relational database, SQLite does not need complex setup procedures, making it an ideal database management system to use for embedded systems and rapid prototyping. In this work, SQLite is set up to generate database, then stored data an easy to use dashboard (Node-Red dashboard) is used to read the acquired data from the data base.

Since the sensor data are constructed data type as MQTT messages that will be published to an MQTT broker, Arduino IDE sketch and Node-Red flow are used to transfer those MQTT messages and then stores them in the database. Node-Red dashboard is a web-based data monitoring tool that can be combined to SQLite.

III. System Testing And Results

The home automation system's proposed design is used to monitor and control the house appliances while maintaining the minimum expected comfortable living conditions. Accordingly, a prototype is shown in Fig. 5 is made to measure the desired variables. This prototype can be located in each room of the house. The sensors acquired data is being processed and used to control smart appliances such as smart LEDs, air conditioners, fans, etc.

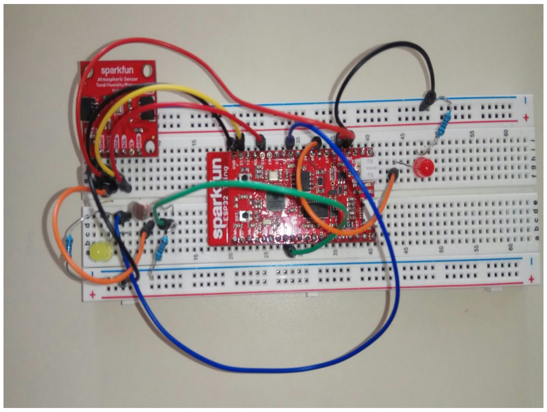

Fig.5. Proposed SCADA system prototype.

In this project a yellow LED is used to test the automatic control ability of the system and a Red LED is implemented to evaluate the supervisory control capability of the proposed SCADA system.

This system monitors light intensity, temperature, humidity, and pressure values. If the recorded light intensity value goes beyond the set high or low value of the threshold level, the controller will generate the signal to turn the LED OFF/ON. Then the user will be notified by the LED status on the dashboard, this feature is presented in Fig. 6 and 7. As shown in Fig. 6, the LDR is darken using a piece of paper to change the state of the LED which is presented in Fig. 7.

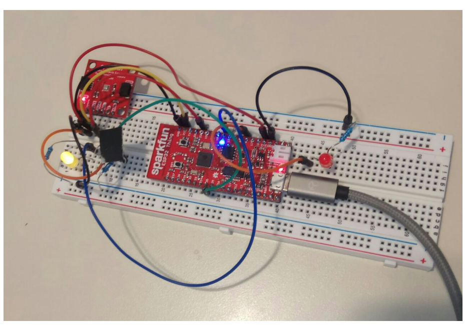

Fig.6. The LED is turned on automatically based on the threshold value.

Fig.7. The status of LED in Node-Red dashboard.

As mentioned earlier, the ESP32 controller subscribes to the published command, created by the end-user through the web interface, and transmitted by the Node-Red. Based on the received message, ESP32 sends a command to the LED to turn it ON/OFF

The implemented system is tested to evaluate its capability. Results provided in Fig. 8, 9 show the recorded values of temperature, humidity, and light density, and pressure, which are acquired and displayed over time through a data-viewing dashboard.

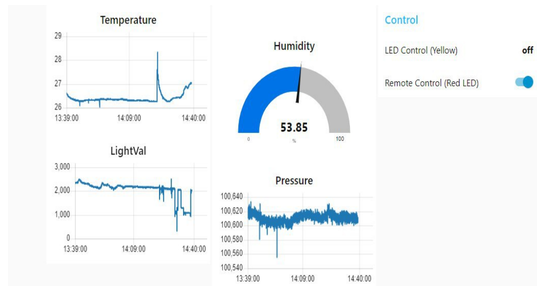

Fig.8. The Red LED is controlled remotely by using Node-Red dashboard.

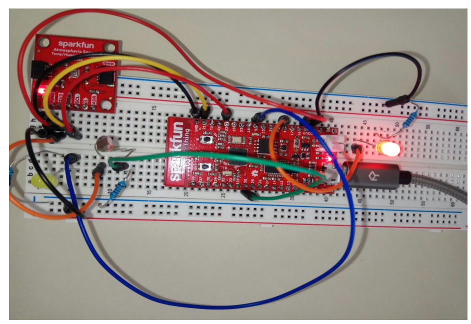

Fig.9. The user remotely turned the Red LED on.

IV. Conclusion

The design of a low-cost and flexible and reliable IoT home automation system is presented. This proposed SCADA system is suitable for monitoring and controlling multi devices at home automatically and remotely. The system works in three phases. In the first one, the system monitors the temperature, pressure, humidity, and light intensity and uploads the data to database storage (SQLite) by using MQTT protocol and Node-Red installed on Raspberry Pi, then to the database connected dashboard. In the next phase, the system automatically controls the LED when the light intensity level exceeds the high or low predefined value of light intensity using a microcontroller (ESP32). In the last one, the supervisory control is achieved; the end-user can monitor and control the home devices through the web interface or mobile application. Once the message is sent through Node-Red along with the MQTT protocol to the microcontroller, it will be executed by ESP32 and turns the devices ON/OFF. The developed system cost is low, simple to operate, and is simply embedded with home devices.

ACKNOWLEDGMENT

The authors would like to thank the School of Graduate Studies, Faculty of Engineering and Applied Science, Memorial University and Pooya Pajouhan Nirouye Golpa Company.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Lamine and H. Abid, "Remote control of domestic equipment from an Android application based on Raspberry Pi card", *IEEE transaction 15th international conference on Sciences and Techniques of Automatic control & computer engineering - STA'2014*, Hammamet, Tunisia, December 21-23, 2014. [Accessed 21 September 2014].

[2] B. S.S. Tejesh and S. Neeraja, "A Smart Home Automation system using IoT and Open Source Hardware", *International Journal of Engineering & Technology*, vol. 7, no. 27, p. 428, 2018. Available: 10.14419/ijet.v7i2.7.10856.

[3] L. Aghenta and M. Iqbal, "Low-Cost, Open Source IoT-Based SCADA System Design Using Thinger.IO and ESP32 Thing", *Electronics*, vol. 8, no. 8, p. 822, 2019. Available: 10.3390/electronics8080822.

[4] A. Sajid, H. Abbas and K. Saleem, "Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges", *IEEE Access*, vol. 4, pp. 1375-1384, 2016. Available: 10.1109/access.2016.2549047.

[5] M. Lekić and G. Gardašević, "IoT sensor integration to Node-RED platform," *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH),* East Sarajevo, 2018, pp. 1-5, doi: 10.1109/INFOTEH.2018.8345544.

[6] P. Rai and M. Rehman, "ESP32 Based Smart Surveillance System," *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2019, pp. 1-3, doi: 10.1109/ICOMET.2019.8673463.

[7] M. Al-Kuwari, A. Ramadan, Y. Ismael, L. Al-Sughair, A. Gastli and M. Benammar, "Smart-home automation using IoT-based sensing and monitoring platform," *2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018)*, Doha, 2018, pp. 1-6, doi: 10.1109/CPE.2018.8372548.

[8] V. Pravalika and Ch. Rajendra Prasad, "Internet of Things Based Home Monitoring and Device Control Using Esp32", *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 58-62, 2019.

[9] R. K. Kodali and S. Soratkal, "MQTT based home automation system using ESP8266," *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Agra, 2016, pp. 1-5, doi: 10.1109/R10-HTC.2016.7906845.

[10] R. K. Kodali and S. Soratkal, "MQTT based home automation system using ESP8266," *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Agra, 2016, pp. 1-5, doi: 10.1109/R10-HTC.2016.7906845.

[11] U. Pujaria, P. Patil, N. Bahadure and M. Asnodkar, "Internet of Things based Integrated Smart Home Automation System", *SSRN Electronic Journal*, 2020. Available: 10.2139/ssrn.3645458.

[12] A. Rajalakshmi and H. Shahnasser, "Internet of Things using Node-Red and alexa," *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, Cairns, QLD, 2017, pp. 1-4, doi: 10.1109/ISCIT.2017.8261194.

[13] K. Chooruang and P. Mangkalakeeree, "Wireless Heart Rate Monitoring System Using MQTT", *Procedia Computer Science*, vol. 86, pp. 160-163, 2016. Available: 10.1016/j.procs.2016.05.045.

[14] D. NAGARAJU, C. KIREET, N. KUMAR and R. JATOTH, "Performance Comparision Of Signal Conditioning Circuits For Light Intensity Measurement", *World Academics Journal of Engineering Sciences*, vol. 01, no. 02, pp. 2007 (1-10), 2014. Available: 10.15449/wjes.2014.2007.