Exploiting Real-time Database for Efficient Electrical Switching

Sheikh Muhammad Farjad
Department of Computer and Information Systems Engineering
NED University of Engineering & Technology
Karachi, Pakistan
smfarjad@outlook.com

***Abstract*—A real-time database system is a type of database system which exhibits the characteristics of real-time processing. Real-time database is one of the innovative ideas for handling the real-time dynamic data which may vary with the passage of time. The rising and falling prices of the stock exchange is an example of real-time dynamic data. In this paper, the idea of exploiting a real-time database for electric switching is proposed along with the different aspects of cost and efficiency. This switching concept is the most primitive idea behind the emerging field of Internet of Things (IoT) based automation.**

***Index Terms*—real-time processing, cloud-hosted database, solid-state switching, smart switches, Internet of Things (IoT), automation**

I. Introduction

The world is getting revolutionized by the wave of the fourth industrial revolution. The rapidly emerging field of the Internet of Things (IoT) has massively affected the world by implementing the technology of automation while making it more feasible as compared to the prior techniques. The industrial sector already relies on robotics for manufacturing and production processes. The domestic sector has also been started being influenced by the wave of automation. The smart home is one of the most beneficial applications of IoT [1]. The IoT is still suffering from the threat of cyber-attacks but when it comes to commodities it has been proven a savior for people living in the 21st century [2].

Other remarkable applications of IoT includes wearable gadgets, smart vehicles, patients monitoring system, smart irrigation, smart education, the intelligent transportation system (ITS), etc. One thing is common in all the applications and this thing is one of the central topics of the paper, that is, real-time processing. Performing the task of real-time processing certainly requires real-time data and the target of storing and managing real-time data is accomplished by a real-time database system. A real-time database system makes the complex operation of real-time processing feasible and it reduces the amount of delay which may often occur during the processing real-time data.

This paper proposes how a real-time database can be exploited efficiently for designing cost-effective and secure smart switches by the integration of a user-friendly interface that can be accessed securely from any location in the world. The practical implementation of this idea is also demonstrated in the paper along with the scope of the system. This must be clear that electric switching is not limited to conventional switchboards. This refers to any electrical circuit which is used for switching ON/OFF. This switch can be operated in isolation or this can be hard-wired with any appliance also. The key idea is automation. Firebase Realtime Database is used for demonstration along with NodeMCU ESP8266.

II. Real-Time Database System

A real-time processing system consists of a real-time database and the characteristics of the data are kept under consideration while integrating the real-time database in the application.

A. Relational Vs. Non-Relational Databases

The application of electric switch depends upon real-time data. A real-time system can be implemented by using SQL (relational database) and NoSQL (non-relational database) also. While selecting the more appropriate architecture of database for the implementation of our project, the following points must be considered:

- *Semantic Interoperability:* The data in the real-time database must be suitable for being used by external agents and applications, this is why it should be easier for a user to understand the database and this can be achieved by unifying formats, data modeling, etc. A non-relational database can handle this issue in a more satisfying way as compared to a relational database [3].
- *Scalability:* Our system must be capable of handling large requests with a low response time. The relational database relies on vertical scalability which is costly and time-consuming because it depends on the hardware and it is impractical due to the hardware limitation. The non-relational database is preferential as it relies on horizontal scalability.
- *Cloud:* Relational database is not an appropriate choice for a cloud-based system because it does not exhibit the characteristic of full content data search. While on contrary, the NoSQL database exhibits all the characteristics which are well-suited for the cloud environment [4].
- *Complexity:* Storing and managing data in tables makes the relational database complex and slower to retrieve.

NoSQL database has the capability of storing unstructured data and this makes NoSQL database suitable for real-time application [4].

B. Firebase Realtime Database

Firebase is a mobile and web application platform that consists of different services provided by Google including authentication, hosting, cloud storage, Machine Learning (ML) kit, cloud functions, real-time database, etc. The service which our application requires is a real-time database. Firebase Realtime Database is a cloud-hosted NoSQL database having the feature of synchronizing data in real-time for all clients connected with it. The clients connected with the same Firebase Realtime Database instance receive updates automatically. If any value in Firebase Realtime Database changes, this modified value becomes accessible for every client possessing the Application Programming Interface (API) key. An API key is selected wisely and it comprises of relatively lengthy string for ensuring the feature of encryption.

III. Hardware Design

For making an interface between the real-time database and switching circuit, a middle hardware component is required. This requirement is fulfilled by using ESP8266. ESP8266 is a cheap and efficient Wi-Fi module which has made the realm of IoT deployment feasible and efficient. For the demonstration, the NodeMCU board is used. NodeMCU is an open-source IoT platform having the firmware of ESP8266 and hardware based on the ESP-12 module. A switching circuit is also required. The proposed system does not use a conventional electromechanical relay (EMR) for switching because EMR is a less efficient solution for electric switching, and it may suffer coil failure or the contacts can get melted due to heavy load and EMR produces arc during the phase of switching. A better alternative of EMR is a solid-state relay.

A. Solid-state Relay Circuit

The reason for preferring solid-state relay over conventional relay is already mentioned in the previous paragraph, a solid-state relay circuit is designed for 220 volts while keeping different safety measures under consideration. A proposed schematic of the circuit can be observed in Fig. 1. This design is compatible with high power appliances also. These high power household appliances include electric motors, heaters, washing machines, etc.

B. TRIAC Specifications

It can be observed in Fig. 1 that the solid-state relay requires TRIAC and optoisolator along with current-limiting resistors and voltage-smoothing capacitors. The prime component of the solid-state relay is TRIAC. TRIAC can be used for switching DC and AC appliances. There are primarily two major families of TRIAC. One includes the BT136 family and the other one includes the BT139 family. The more suitable for high power electric appliances is the BT139 series.

C. Safety Components in Circuit Design

In order to isolate low voltage components from high voltage load, an optoisolator is used which isolates the lower voltage part of the circuit from high voltage load. This protects the ESP8266 from getting burnt. The suitable and more appropriate optoisolator for our design is opto-triac. The proposed schematic contains the MOC3021 opto-triac.

TRIAC-based circuits often suffer from Rate Effect which arises due to the switching noise or transient in the AC line. In order to tackle this problem, the snubber circuit is connected in parallel with the TRIAC. This snubber circuit consists of a capacitor and resistor connected in series. This snubber circuit does not let the TRIAC divert from its function by suppressing the surges.

After parsing all these steps, an optimized TRIAC based switching circuit is designed by following the schematic diagram of the circuit shown in Fig. 1. This switching circuit is adequately compatible with the load of 220 volts. This switching circuit is more efficient in terms of power loss and cost also.

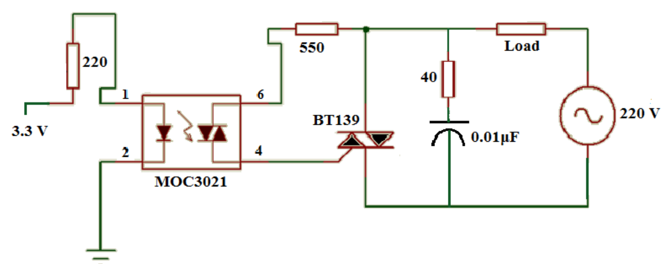

Fig. 1. Circuit Diagram for TRIAC-based Switching

IV. System Architecture and Prototype

After elucidating the main units of the system, it is time for demonstrating the implementation. The architecture of the real-time database based electric switch consists of the following components:

1) *User Interface:* The user interface (UI) consists of a web application or a mobile application that is integrated with the Firebase Realtime Database SDKs. The Firebase Realtime Database is linked with the UI through an API key. The main functionality of the UI is to provide the user with a user-friendly interface for switching any appliance and link the user with Firebase Realtime Database. The UI can be further customized according to the need. The mobile application for the demonstration includes the features of the toggle button and voice command which means you can perform the task of switching by toggle button and voice also. The mobile app is shown in Fig. 2.
2) *Firebase Realtime Database:* The second and most essential component is Firebase Realtime Database. The user sends the command of turning the switch ON or

OFF through the mobile application. This mobile application is linked with the Firebase Realtime Database and this application interprets the command entered by the user and changes the corresponding value in the real-time database of Firebase. The most simple and adequate value for the representation of the switching state is a bit. 0 represents ON and 1 represents OFF. If the user enters the command of turning the switch ON, the 'switch' component in Firebase Realtime Database contains the value of "0" because 0 represents the ON state. In case of turning off the switch (through the application), the same component of the real-time database holds the value of "1".

3) *NodeMCU Board:* NodeMCU Board is an open-source IoT(Internet of Things) platform for building the ESP8266 based projects. Its firmware runs on ESP8266 WiFi SoC (system-on-chip). NodeMCU board is a programmable board which can be used for performing different tasks employing WiFi. This is compatible with Firebase Realtime Database as it runs on the Arduino IDE and it contains the Firebase library which can be used for integrating Firebase Realtime Database with the NodeMCU board. The NodeMCU keeps checking the value of the required component in the Firebase Realtime Database. If the required component in the database contains "0" it is interpreted as ON and then the NodeMCU sends the HIGH signal to the corresponding pin which is attached with the switching circuitry. Similarly, in case of switching OFF the same pin will be at the LOW signal.

4) *Switching Circuitry:* The GPIO (General Purpose Input Output) pins of NodeMCU provide the DC value of approximately 3.3 volts. This signal is fed into the input of the switching circuit consisting of a solid-state relay mechanism. TRIAC is used for switching mechanism while optocoupler is to ascertain the protection of the circuitry by isolating the AC (alternating current) component from DC (direct current) component. This increases the stability and lifetime of the switching circuitry. A sample circuit is shown in Fig. 1.

The aforementioned four stages make it possible for users to turn the switch ON or OFF while being in any part of the world. The users need to interface with a mobile or website application for sending the command. The architecture for electrical switching based on the real-time database is illustrated in Fig. 3. It is worth mentioning that the label 'SWITCHES' in Fig. 3 refers to the solid-state switching circuit.

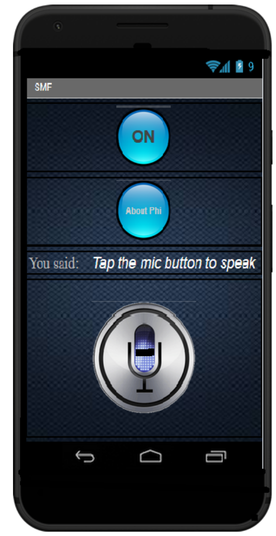

Fig. 2. Mobile Application for Switching Interface

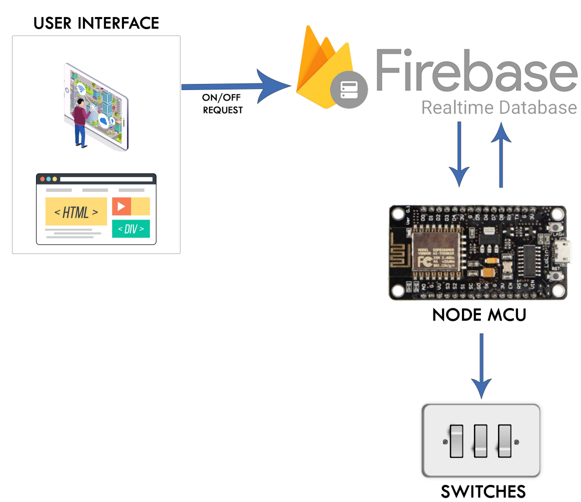

Fig. 3. Architecture for Real-time Database Based Switching

V. Related Work

The idea of automation is not new to the world. Many researchers have already proposed different implementations of automation systems using IoT. Mandula et al. proposed the idea of mobile-based home automation using the Arduino board which relies on Bluetooth for indoor environment and Ethernet for the outdoor environment [5]. Bai et al. penned down a similar idea by using a four-quadrant user interface and this proposed design has the feature of monitoring the power consumption also [6]. Benazzouz et al. discussed the idea of integrating IoT by distributing and hosting the applications in a cloud which makes access more secure [7]. Most of these proposed systems rely on Bluetooth for an in-door environment and Ethernet for the out-door environment. ESP8266 low-power and cheap Wi-Fi SoC (system-on-chip) have shifted the IoT paradigm to the next level of advancement. This SoC is a cost-effective, feasible, and more reliable solution for integrating IoT on small and large scale production.

This paper further optimizes the IoT based automation system by embedding it with the features of the real-time database, ESP8266 low-power Wi-Fi SoC, and a feasible design of electric switching circuit using solid-state technology. These three features are integrated with a user-friendly UI which has the function of voice command and this voice command can be taken from the user in any native language by integrating the mobile application with a suitable API.

VI. Conclusion & Future Scope

In this paper, we proposed how the real-time database can be utilized for implementing a feasible, cost-effective, and stable IoT based system. This paper also highlighted the effective hardware interface with the real-time database and proposed the optimized solution for switching circuitry which can be implemented for the long-term system. This proposed system is cheap which makes it affordable for the people to implement in their home automation.

The solid-state technology makes the switch compatible with some high power appliances as well. The switch can be connected or hard-wired with any sort of household appliances for controlling the switching mechanism remotely from any place. There is no need to deploy separate components for in-door and out-door environments and this is also another advantage that reduces the size of the switching circuitry. The proposed system can be utilized for further applications in the future. It can be used as a sub-system in large-scale production. This switch can be integrated with any electric appliance at the time of manufacturing the targeted appliance.

References

[1] S. Dey, A. Roy, and S. Das, "Home Automation Using Internet of Thing," 7th IEEE Annual Ubiquitous Computing Electronics & Mobile Communication Conference, 2016.

[2] J. H. Han, Y. Jeon, and J. Kim, "Security considerations for secure and trustworthy smart home system in the IoT environment," 2015 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1116–1118, 2015.

[3] V. Abramova, J. Bernardino, and Pedro Furtado, "Experimental evaluation of nosql databases," International Journal of Database Management Systems, vol. 6, no. 3, pp. 1, 2014.

[4] M. A. Mohamed, O. G. Altrafi, and M. O. Ismail, "Relational vs. NoSQL Databases: A Survey," International Journal of Computer and Information Technology, vol. 3, pp. 598–601, 2014.

[5] K. Mandula, R. Parupalli, C. H. A. S. Murty, E. Magesh, and R. Lunagariya, "Mobile based home automation using Internet of Things (IoT)," 2015 International Conference on Control Instrumentation Communication and Computational Technologies (ICCICCT), pp. 340–343, 18–19 Dec. 2015.

[6] Y. W. Bai, H. W. Su, and W. C. Hsu, "Indoor and remote controls and management of home appliances by a smartphone with a four-quadrant user interface," IEEE International Conference on Consumer Electronics (ICCE), 8-10 Jan. 2017.

[7] Y. Benazzouz, C. Munilla, O. Gunalp, M. Gallissot, and L. Gurgen, "Sharing User IoT Devices in the Cloud," in IEEE World Forum on Internet of Things (WF-IoT), IEEE, pp. 373–374, Mar. 2014.

Sheikh Muhammad Farjad is an undergrad student of computer and information systems engineering at NED University of Engineering & Technology. His research interests include Internet of Things, Artificial Intelligence, and cybersecurity. He has served as a Research Student at Research Center for Artificial Intelligence (RCAI). He has worked on many projects including Wi-Fi switch, smart irrigation database system, automated water pump, and home automation system.