Development of Smart Earth Leakage Circuit Breaker Using IoT and Power Electronics

Dr Rashesh Mehta 1, Junaid Saiyad 1, Niyatee Smart 1, Gordhan Mali 1, Dr Vithal Kamat 2

1 Department of Electrical Engineering, Birla Vishwakarma Mahavidyalaya, Anand
2 Baroda Meters Ltd., GIDC, Vithal Udyognagar, Anand.

E-mail: rpmeha@bvmengineering.ac.in

Abstract. The Smart Earth Leakage Circuit Breaker research is based on IoT & power electronics-based protective switchgear for domestic consumers for the protection against earth leakage faults. In this paper, we have shown how to create a custom web server using HTML and CSS language and hosting this webserver to ESP module using Serial Peripheral Flash File System (SPIFFS). Along with protection against earth fault it provides, controlling of tripping current or fault current, monitoring voltage, power, current, power factor parameters of the connected load, monitoring fault parameters, indicating the device status whether the ELCB is on or off, and the condition of ELCB it is in faulty condition or healthy condition and to let the user set the value of the tripping current on the webserver at which the ELCB must trip.

Keywords: Smart home, IoT based, Earth Leakage Circuit Breaker, Webserver, Smart Energy Systems, Automation, Protection

1. Introduction

Earth Leakage Circuit Breakers (ELCB) are very much essential for the safety of human beings in residential as well as industrial premises. The significance of ELCB for home installations is elaborated in [1]. This is the era of technological revolution and the “Internet of Things” (IoT), so using IoT we can make our existing devices smarter and better with advanced features and enhanced protection that is what Smart ELCB is all about. It will be a smarter device and user-friendly compared to an existing system and one can easily operate it[2].

This paper is focused on building a unit of Smart Earth Leakage Circuit Breaker (Smart ELCB). This ELCB will show the status of the ELCB device whether it is on or off, ELCB will provide the tripping function when the earth leakage fault is detected, it will also monitor voltage current and power on a web server or our monitoring medium. By using the parameters measured by the Wi-Fi module the limits of the tripping current can be set in mA, which will be measured continuously as long as the control circuit is powered on. This control circuit takes power from the SMPS circuit which converts 230 V to 5 V DC and provides this supply to the ESP module and energy metering chip to measure the energy and control parameters[2]. These parameters are shown on the webserver using the “GET POST” method for the asynchronous web server.

The earth leakage current beyond 30mA causes the death of the electrocuted human. Although the range from 10mA-30mA has a fatal effect on people's life. So the sensitivity of such currents is
necessary for safety in domestic applications wherever the person may approach in direct contact through the electric device in places like schools, labs, workshops [4] etc.

Smart ELCB is one of its kind protective switchgear for domestic purposes which not only protects the users in the earth fault condition but also provides satisfactory operation and remote control to the user. The device can be operated remotely within the coverage area of the local area network (LAN) provided by the ESP8266 based WiFi module. The advantage and use of this ELCB exceed the capability of what the current ELCB provides. Ratings of this ELCB are 16A, operating range, and 10-30mA fault current tripping. The main feature is dynamic control of ELCB operation like the selection of fault current, and remote control of ELCB.

We have proposed a new design of Smart Earth leakage Circuit Breaker (ELCB) which is can provide better protection. It has other modern features like “smart” sensing, controlling, and monitoring of the voltage, current and status. It has the capacity of online connection through a Wi-Fi network and allows remote operation and control. With the help of IoT, the new features achieved are real-time data uploading and refreshing with the use of AJAX scripts, HTML elements to control the actual physical value like mA current and controlling the entire ELCB device. By using the Power Electronics technology for switching action, high-speed disconnection and automatic reconnection are implemented.

The paper is presented in sections as explained here. The outline and the blockdiagram are explained in section 2. Section 3 depicts the implementation methodology is explained. Section 4 consists of results and discussion. The conclusion is given in section 5.

2. The Overview of the Proposed Smart ELCB

The Block diagram of the smart ELCB module is shown in figure 1. The Components, connections, and details about the working of each component are listed below.

![Block diagram of Smart ELCB](image)

Figure 1. Block diagram of Smart ELCB

Following are the working of the components used in the smart ELCB module.

- **MCB:** - Miniature circuit breaker used with ELCB to protect not only from earth leakage currents but also from the short circuit.
- **Triac:** - It is used for switching purposes, It is a three Quadrant Triac which is controlled by MOC3021 Optocoupler and ESP8266 microcontroller. The port pin sends the gate signal so the Triac is fired and the load is turned on, It is controlled by online web server user.
SMPS: - The power supply circuit for the control circuit to enable the controlling of the ELCB takes the mains supply directly to keep the control circuit turned on to detect whether the device is on or off.

ESP8266: - It is the heart of the control circuit. The monitoring of energy parameters, control parameters, remote controlling, and hosting of the webservers, is achieved through this device only.

HLW8012: - It is an energy monitoring chip used for the measurement of energy parameters, which gives the output RMS value of the current, voltage, power and power factor of the connected load.

Optocoupler: - It is used for providing isolation between Triac and ESP8266. Also, it delivers the signal from ESP8266 to Triac to operate it safely.

Core Balance Current Transformer (CBCT): - whenever the fault appears the CBCT secondary provides a proportional signal for the detection and measurement of fault.

To measure the amount of fault current flowing from primary we are connecting the CT secondary terminals to the A0 and GND which will read the proportional values of the fault current.

3. Implementation Methodology

The Smart ELCB module discussed in this paper works on the Asynchronous web server through which the controlling, monitoring and switching of the load are done. The web server is hosted by the ESP8266 module which is used in the control circuit of the ELCB[3].

The ELCB also measures the fault current by the use of CBCT secondary terminals and it is calculated by the ESP8266 module. This module calls the required function whenever the Client Request to the server is given. By the GET POST method, the Values calculated by the HLW energy metering chip and CBCT sensors are provided on the webserver for monitoring purposes.

3.1 IoT based communication

Here, we are using the ESP8266 WiFi module as the medium for our IoT communication. This module will work as a controller of the device as well as a communication medium to communicate with the webserver by which we are operating the ELCB and connected loads can be controlled simultaneously[9]. So basically, we are using IoT communication for operating, controlling, and monitoring ELCB so we can easily handle our device[3].

![Figure 2. Physical Demonstration of IoT based smart ELCB](image)

The above image in figure 2 shows the physical demo of smart ELCB, to study IoT based communication we must understand the basic components used in it as per the block diagram the circuit contains all the above-mentioned components of ELCB although, the main component used is ESP8266 asynchronous web server library and ESP async TCP library to setup the asynchronous webserver.

3.2 Creation of web server and its application

The asynchronous web server from the library provided by Random Tutorials here ESP async library is used. As we have used this we can connect more than one connection at the same time. To use this library just include this library with other important files to be included for the webserver like File system library FS.h. This library will help us to serve the HTML indexes and the CSS file for desired sheet styling as per our web server. Then ESP async TCP library is utilized for arranging the data in
such a fashion which allows the safe exchange of data between the server and client without disturbing or leaking the data [3]. After using these libraries we can successfully set up the webservice on our ESP module.

First, we have to select the port for the webservice to which the IP address will redirect the user, and the HTML, CSS, and JS file will be served at the same port for which the webservice is hosted this can be done by command:

```c
AsyncWebServer server(80);
```

We have to give the requests for giving the input to the web server as shown in the image in figure 3. These requests are called request handlers. The first request is to provide the index file for the webservice which is an HTML file, then we are adding CSS files and corresponding requests for images, data are given to the web server these requests provide required data to be shown on the webserver for real-time monitoring of parameters and these parameters are taken from the sensor nodes explained further in this paper. Finally, after the creation of the backend process of getting data, these data will be available to the remote web server and hosted on the ESP module whichever device will connect with these ESP will be able to access this web server on a particular IP address.

The server.on request is required to communicate with the webservice by using a microcontroller, the request contains “/URL to be redirected” for the execution of a particular function in that URL, a method to get the data “HTTP_GET”, and a server request function “AsyncwebServerRequest*Request” which serves the purpose of accessing the data at that particular URL. Similarly, Request handlers use SPIFFS to get the HTML, CSS file as per figure 3.

**3.2.1 Use of serial peripheral flash file interface:** The serial peripheral interface flash file system (SPIFFS) is used to serve the HTML and CSS file to the webserver by the request handler. These files are contained in the flash memory of the ESP8266 chip and are used whenever the requested function value is required by the webserver for example if the HTML index is required by IP address to redirect it to the HTML file then this HTML file will be included in the flash memory of chip and the insertion of these outer files to the flash memory of the ESP8266 will be done using the FS.h library, this is not enough to work along with library we will need to use the spiffs installer for the ESP8266 which can be obtained from here [spiffs installer]. Once this installer is installed, sketch data is uploaded from the particular data folder which contains necessary files, icons, and other files to use for the webserver.
3.3 Data monitoring to the webserver:
We send the measured parameter values of the energy meter to the web server using HTML, JAVASCRIPT, and REQUEST HANDLERS [2]. The HTML code stores the processor values which are updated by the Sensor values or parameters and then by the use of javascript we are calling the function to give the output on that particular processor of data to be stored and finally when we send the request handler to the Async web server it gives us the particular value of the parameter on the webserver[9].

![Diagram](image)

**Figure 4.** Data transfer from sensor pin to Webserver

As explained in the above figure 4, the data from the sensor is given to the variable processors that are used in the main program as shown below and then it is linked to the HTML code with the help of the same processor functions and when the value is being extracted the same processor gives out the Value of sensors and the HTML web server is updated with new values.

```java
String getvoltage() { //This is the function to be called
floats = Hlw8012.getvoltage(); //getting the voltage parameter and storing the value of voltage in the
volts variable which is of float data type.
return string (volts); //This function will return the value of volts i.e the voltage parameter
}
```

**Figure 5.** Voltage measurement function for placeholder

The function “getvoltage ()” as shown in figure 5 gets the RMS value of voltages from the energy meter chip with the help of HLW8012.get voltage function. It returns the output in string form to display on the webserver, the function “getvoltage ()” is also a string, this is because the use of placeholder value in HTML file is done as string form only, that’s why each parameter is considered as a string.

```java
String processor (const string & var) { //This is the processor variable
if (var == "voltage") //if variable var is equal to voltage placeholder inside the html
return getvoltage(); //then return the value of voltage.
}
```

**Figure 6.** Voltage value to HTML placeholder

```javascript
server.on("/voltage", HTTP_GET, {AsyncWebserverRequest request] { //server.on call which will redirect to voltage placeholder
url then we use HTTP_GET method to get the value of asyncwebserver request
request.send(200, "text/plain", getvoltage([_sh]],
); //Sending the request to the above "voltage" url by calling the function which returns the value of voltage parameter
```

**Figure 7.** Requesting to get data on the server
So as we have shown above 3 parts of code in figures 5 to 7, to send the sensor value from sensor to webserver, here we have shown just one value of voltage parameter. First, we are getting the voltage parameter by using HLW8012.getvoltage for the HLW8012 sensor which will give us the RMS voltage value. Then we are storing that function's output, which is the actual voltage value, into the place holder created in HTML where the voltage value will be given to HTML id and used as particular data for HTML file for webserver, and then finally we are requesting to send the data to the webserver by get method as shown above in figure 7.

3.3.1 HLW8012 Energy Monitoring Chip: - To choose the compact, accurate, reliable, and effective energy metering chip to be used in smart home devices and automation devices this chip HLW8012 is considered as the best option [11]. This chip interfaced between power mains and the connected load, and according to the block diagram, it is used after the Triac of smart ELCB, so the connected device will only show the readings whenever the Triac is turned on and after that only the chip will give the output, the pin diagram of the chip is shown in figure 8 [6].

![Figure 8. HLW8012 pin diagram](image)

The chip gives out square wave pulses proportional to the main power input, voltage, and current. The datasheet of the chip contains information about the chip output measurement as the frequency function, like for 12W active power the output pin frequency is 1 Hz and for computation of voltage and current, the SEL pin is regulated high and low, if the SEL pin is high the voltage output will be observed from the CF pin of the chip and if the SEL pin is low current output will be observed. For 0.5V or 15mA 1Hz, proportional frequency is generated. With the use of these data along with the modified calculation and programming this chip can be easily used for energy monitoring, the basic code for the working of this chip is from HLW8012code this repository.

By the use of the computation and data processing from this chip, as shown above, the energy parameters are sent to the webserver.

3.3.2 Final data monitoring on the webserver: - The following image in figure 9 shows how the data monitoring and controlling is working on the actual web server that we’ve been implemented as in previous sections. As we can see there are the customization index logos and tables arranged and editable HTML file used with CSS file which improved the visibility of the custom webserver. Actual
data upload during the snapshot can also be shown in which energy parameters and control parameters show the real-time data of the load. The tripping current limit is set to 30mA and the ELCB state is on hence condition is “NO FAULT”. Also, some fault current value is shown, the “maximum fault current” is the value of the leakage current flowing during the entire operation of the smart ELCB.

Figure 9. Web server during operation

3.4 Data Updating to the webserver

To get the real-time data from the sensors continuously and Serving them on the webserver for monitoring purposes we need to use some Data updating techniques to continuously monitor the real-time data from the sensor readings which shows the accurate operation of the device in terms of reliability and use of it[3]. As we have used a customized webserver we can use two techniques for data refreshing to web server[8].

Adding HTML script to reload the page: - In this method, the meta request method is used to reload the whole page after a given time the meta tag is used as shown in figure 10, but the limitation of this method is that while reloading the webpage or web server, each element of the HTML web server is updated as new and it may take time due to loading of elements which may mess up the data and customization. The main problem was with HTML button elements, We have used ON and OFF buttons on the webserver whenever the button is pressed the action takes place, but when the page is reloaded the button automatically resets which was useless for real usage of this method and also this is the main reason to not using this method for data updating. Although it works fine if we have only Data string monitoring of the webserver.

Figure 10. HTML page reloading script

Using Asynchronous Java XML script: - In this method, JavaScript is used for updating each parameter, the main advantage of using this method is that a particular data set is updated, unlike the whole web page. This method uses a placeholder value which is provided by the ESP8266 code and when the given variable is provided with the value, the value stores it as a placeholder, This JavaScript requests a particular placeholder to get the latest reading from the sensor nodes, when it receives the value it updates HTML element with that placeholder id.

The image in figure 11 describes the data updating of the fault current value. As we can see the “maxfaultcurrent” placeholder is used to get the fault current value as well as updating the value. At the start, the function is defined for a particular id to store and update the value. The HTTP request for
taking the latest data is made and after receiving the latest fault current value it is sent to the HTML id which will be the updated value of fault current. And at last, the amount of time is defined for how many milliseconds the value should be updated. Here 2000 i.e. every 2 seconds the latest value of fault current will be shown on the webserver. The same method is used for other parameters also to get the real-time data monitoring parameters.

![Javascript to reload maxfaultcurrent data](image)

**Figure 11. Javascript to reload maxfaultcurrent data**

### 3.4.1 Customization of webserver:
- Data accumulated by the webserver needs to be organized properly to make it easily understandable and easy to read. The web server used in this image contains logos, Custom styling sheets, various data tables, and parameters to display the actual reading of the particular load devices. The Parameters are divided into two tables energy and control parameters to show corresponding values of the connected load. This whole process is made possible due to table formatting, CSS, HTML elements and spiffs to interface these outer files and logos to the flash memory of the ESP8266.

### 3.4.2 Fault current calculation:
- Here, the fault current is measured from the secondary side of the CBCT. So, whenever the earth leakage occurs, we will get a proportional signal on the secondary side, and that we will use it as our input data to the analogue pin of the microcontroller, this signal will be very small in magnitude in terms of 40-50mV and it will be in the range of mA. Depending upon the connected burden resistor across Core balance CT terminals, the proportionality is obtained by standardizing this values with the help of multi-meter and then simply selecting the particular multiplier for accurate measurement of the fault current flowing in faulty as well as healthy condition[7].

![Fault current data](image)

**Figure 12. Fault current data**

The CBCT secondary terminal gives out the signal proportional to the fault current flowing through primary the samples are taken and measured in the microcontroller (ESP) and calibration is done. Samples are taken when the measurement function is called in the main loop and the value of voltage is obtained by multiplying these samples with the appropriate multiplier value we get a range of the measured current value calibrated by the multimeter[7]. This method increases the processor time, through its effect is not seen visible but memory consumption increases and some modification should be done to have valid fault current calculation code. After this process the correct and accurate
value of fault current is obtained and by use of HTML and JavaScript, we are sending these values to the webserver. The webserver layout is shown in figure 12.

3.5 Controlling Of Tripping Current Value
We are using the HTML slider using input as a range which will add the slider on our web server. This slider is storing the integer values from minimum limit 10 to maximum limit 30. By using the if loop to constantly monitor if the value of measured fault current is greater than the slider value then the ELCB switch must be stop hence tripping the load, if the value of fault current is smaller than that slider value the ELCB must auto reclose and the power is restored[8].

![Tripping in (mA)](image)

**Figure 13.** Earth Leakage Current Value Set Slider

As shown in figure 13, the method gives us the control to limit the tripping current value that when the ELCB should trip. Hence giving us control over the selection of fault current at which, ELCB should trip.

The slider value variables were not saved inside the HTML code due to which there was no update in the slider value on the webserver. After saving the values inside the parameter and updating the input message variable we converted the string input from the webserver to the int value after which we can successfully detect the tripping value and trip at a set value of current.

3.6 Switching Control
To control the switching of the ELCB, We have demonstrated the use of HTTP requests made by HTML elements which control the execution of a particular function when called upon, by this requests the webserver is redirected to the function containing the actual function to operate, hence in our case we have used the buttons elements and when these buttons are clicked on the client-side webserver the HTTP requests are made and the function inside this request that is to turn on or off the particular port of the ESP8266 microcontroller and when this port is controlled operation of Triac is controlled hence we can use this method for remote switching application of the smart ELCB [10]. Below, we have shown detailed components and usages to make this process more understandable.

3.7 TRIAC AS A SWITCHING DEVICE.
Various advantages of thyristor and TRIAC are explained in [5]. Because of the above all merits, we had chosen to use an electronic switch (TRIAC) rather than the mechanical component of ELCB. We disassembled the component from ELCB and connected TRIAC to it.

Bi-directional switching devices like TRIACs are used for switching functions in AC circuits. A positive voltage or a negative voltage can be used to trigger the TRIAC in the direction required. Thus, four different zones are possible for the operation of a TRIAC. At the starting of our project, we operated it on the 1st and 3rd quadrant but now we give MT2 the same supply to the gate because it is sinusoidal so, in other words, we used natural commutation for triggering our gate[6].

3.7.1 Direct ESP with Triac: - The connection is useful to use the Triac controlling device without any interface component to reduce the overall cost of the component as well as hardware complexity. The technique uses the knowledge of basic electronics to connect the active connection to the ground and using Triac in various modes of connection as per the 3 quadrant operation of the Triac.
To start the Triac we have to give a gate signal then only the Triac will turn on so here we are giving gate signals from our microcontroller ESP8266 through 560Ω resistor to the Triac and controlling the operation of the Triac by controlling the signal of the gate. The circuit for this is shown in figure 14. The neutral terminal here is taken as a reference to 5V and the gate supply is given through D6 pin and GND so that the Triac will operate in the 2nd and 3rd quadrant due to signal obtained during the actual LOW in the Port pin. The port pin now acts as active high.

3.7.2 ESP with Triac using Optocoupler: - The Optocoupler is required mainly to interface between power and control circuit but in our case, we have to use it to protect the differentiation between reference terminals. The Optocoupler used is of MOC3021 series with diode main current rating 1.5V, and optical Triac ratings of 600V, 4A.

As we are connecting our Triac to 230-240V supply and this high voltage will be dangerous to our controller so it is necessary to isolate our controller from this amount of voltage so here we are using an Optocoupler to provide isolation to our controller so it won’t get damaged due to high amount of current or voltage. You can see the connections of that in figure 15 [12]. More details about Optocoupler used in the smart ELCB is discussed in further points.
3.7.3 **OPTO-COUPLER MOC3021**: A TRIAC can be used to drive the Optocoupler MOC3021 which has a zero-crossing detection capacity. The loads up to 400 V can be driven by the output driven by a TRIAC. As TRIAC can conduct in both directions, AC loads can be easily controlled. The TRIAC will conduct as soon as the AC wave crosses 0 V. This helps in avoiding the injection of direct peak voltage on the load and thus damage is prevented. The sufficient rise and fall times help in controlling the output voltage. Thus, MOC3021 is a perfect choice to control the high voltage AC loads. The pindiaagram of MOC3021 is shown in figure 16.

![Figure 16. Moc3021 Optocoupler Pinout](image)

As the current rating of the Optocoupler is limited, it cannot drive loads directly. A power electronics-based switch, TRIAC is connected at the output. A TRIAC provides sufficient current required to drive the load. TRIAC is controlled by the Optocoupler. Figure 17 shows a circuit diagram wherein an AC bulb is controlled by a microcontroller along with an Optocoupler and TRIAC. The connection with the main load and the Triac is similar but the difference is of HLW chip in case of smart ELCB as per block diagram, the energy metering chip is interfaced between load and Triac else the connection is same as below, this connection can be used as a reference for use of Triac with AC load control with other loads also[13].

![Figure 17. Connecting Moc3021 Between Mcu And Triac](image)

4 **Result and Discussions**

This section shows the final results and operational condition of the smart earth leakage circuit breaker as we can see there is an off and on condition of ELCB in both normal and fault conditions, the webservice snapshots are shown.

4.1 **Normal OFF condition**

When ELCB is in OFF condition we can see in figure 18 that the parameters on the load side are zero and the state of ELCB is also OFF and there is no fault condition because ELCB is in a normal OFF condition. And fault parameters will also be zero.
4.2 *On Condition* *(No-Fault operating condition)*

When ELCB is in ON condition we can see in figure 19 that the parameters on the load side are as per the load connected on the load side and the state of ELCB is showing as ON. There is no fault condition because ELCB is in a normal operating condition. And fault parameters will also be zero because it is in no-fault operating condition[7].

4.3 *Earth leakage Fault occurred*

Figure 18. The web server in the normal OFF condition

Figure 19. The web server in the normal ON condition

Figure 20. The web server in a fault condition
When there is an increment in earth leakage current greater than 30mA the ELCB will turn OFF. Then ELCB is in fault condition we can see in figure 20. The parameters on the load side are zero because due to fault ELCB is trip and is in OFF condition and the state of ELCB is showing as OFF. And here we can see the value of fault parameters fault current and amount of maximum fault current.

4.4 After Fault clearance

Figure 21. Web server auto-reclosed

After the tripping of ELCB one will clear the fault as the fault will be cleared ELCB will be turned ON to normal operating condition automatically. We will be able to see the load side parameter again as per the load connected and the value of the current fault current is zero and the last maximum value of fault current will be stored which we can see in figure 21.

5. Conclusion

In this paper, the development of smart earth leakage circuit breakers for household applications is explained. The implementation is based on IoT, webserver interfacing, and parameter measuring to meet Smart ELCB features. The prototype is already developed and tested.

This ELCB contains an automated power restoration feature when the fault occurs. Further, it includes the feature of energy and control parameter monitoring. In which we can monitor or can see the values of load energy parameters like voltage, current, power, etc., and in control parameters, we can see the value of fault current and the maximum current value of which ELCB has trip last. And setting the value of fault current at which the circuit breaker must trip. And the range of this tripping value is from 10 mA to 30 mA.

For future work, we can make this experimental model of ELCB to the Domestic product, and by manufacturing; we can establish this device for housing protection application and commercialization.

References

[1] A P Purnomoadi et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1098 042047
[2] Zorzi, M. Gluhak, A., Lange S, & Bassi. A, (2010). From Today’s Intranet of things to a future internet of things: a wireless and mobility-related view. IEEE Wireless Communications, vol:17, no(6), 2010.
[3] Castellani A, P Gheda, M. Bui, N Rossi, M & Zorzi, M. (2011, June). Web services for the future internet of things: a wireless and mobility-related view. IEEE Wireless Communications, vol:17, no(6), 2010.
[4] Wolff H and T.G.F Atherton, “Miniature circuit-breakers”, Journal of the Institution of Electrical Engineers, Vol. 1, 1955:4:232-234
[5] Jicheng Yu, Yingying Tang, and George G. Karady, “Design of an IGBT-based electronic circuit breaker”, IEEE North American Power Symposium (NAPS), 2010.
[6] Abirami P. and Merin Lizbeth George, “Electronic circuit breaker for overload protection”, IEEE International
Conference on Computation of Power, Energy Information and Communication (ICCPEIC), 2016.
[7] Vasavada M. R., Patel V.S. and Prajapati J.R, "Development of Intelligent Automatic Electronic MCB and ELCB Using Fault Diagnosis Technique," 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), 2020, page no. 346-350.
[8] AbdAzzis Z.H, Noor Mohd N, and T. Ibrahim, "Automated Electrical Protection System for domestic application," 2013 IEEE 7th International Power Engineering and Optimization Conference (PEOCO), 2013, pp. 23-28.
[9] Jose, A. M. Varghese, F. E. Jenson, P. Jacob, D. M. Anumod and D. Thomas, “Solid-State Circuit Breaker based Smart Distribution Board with IoT Integration,” 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 54-61, 2020.
[10] Ma L, Li Z and Zheng M, "A Research on IoT Based Smart Home," 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp. 120-122, 2019.
[11] Datasheet for HLW8012 from “http://www.hiliwi.com/product/1.HTML”
[12] Datasheet of Triac from “https://datasheetspdf.com/pdf-file/583759/NXP/BTA316x/1.”
[13] Datasheet of Optocoupler from “https://pdf1.alldatasheet.com/datasheet-pdf/view/53870/FAIRCHILD/MOC3021.HTML”