

## **WEB CONTROLLED SMART SWITCH**

JagritiGupta<sup>1</sup>, EktaHooda<sup>1</sup>, Antim Dev Mishra<sup>2</sup>, Bindu Thakral<sup>3</sup>

<sup>1</sup>Student, Ansal University, Gurugram-122003, Haryana, India

<sup>2</sup>Sr. Technical Officer, Ansal University, Gurugram-122003, Haryana, India

<sup>3</sup>Assistant Professor, Ansal University, Gurugram-122003, Haryana, India

**Abstract:** A Smart switch is the most basic building block of a home automation system. Web controlled smart switches offers an accessibility of controlling any electrical appliance, using the smart phone or web browser, with the connectivity capabilities of Raspberry Pi 3 development kit. Such Smart switches ease our toiling life and aim to catch up with fast growing metropolitan life. This paper presents the design and implementation of Web controlled smart switch using a Raspberry Pi3 development kit and Relay module employing a TOR web browser. A smart switch for any electrical appliance can be controlled from anywhere in the world as long as there is an internet connection.

**Keywords:** Raspberry Pi, IoT, Relay Module, TOR, Node.js

### **1. Introduction**

Home automation is a growing industry; using IoT enabled devices interconnected with each other, making our day to day tasks much easier. By smartly using electrical appliances, home automation also conserves energy.

A smart switch is the most basic building block of a home automation system. Raspberry Pi model 3 has inbuilt WIFI and Bluetooth connectivity which makes it the perfect board for IoT based projects. By using a Relay module, the board can control almost any electrical appliance.

People traveling or commuting daily, often forget to check their electrical connections before leaving and have no way to connect or disconnect them. Only until, they reach home, do they realize the damage caused. [1]

By making a smart switch, any electrical appliance can be controlled from anywhere in the world as long as there is an internet connection.

IOT projects are made possible with both electronics and communication knowledge. Using a Raspberry Pi 3 development board and relay module, requires basic knowledge of electronics and electrical connection. Interfacing and programming will also require basic knowledge of Raspberry Pi's operating system and a few high level computing languages.

Extending any connections which interacts directly with a home network and its appliances exposes many security risks. Using a TOR based connection, bypasses this issue.

### **2. Literature Review**

The widespread opinion of a smart switch is that this type of devices will surpass any other form of intelligent computing and communication in a very short time, acting as a useful facilitator for the internet of things. Although smart switches may seem to be a new invention, they have been in use for a relatively long time.

Nikolas Tesla designed the first remote control in 1898 to operate a toy boat. He achieved this using radio waves to send instructions from a handy device to his boat. [2]

Several years later, Jim Sutherland invented the Electronic Computing Home Operator (ECHO) that could be used to store recipes, control a home's temperature, toggle appliances and relay messages.[3]

R.Morale et.al designed a smart switch to connect and disconnect electrical devices at home using Internet. Smart Switch was connected to internet using Wi-Fi, to a computer, Smartphone, tablet or any device with internet access. The smart switch can be controlled simply by sending a number one or a number zero to switch the electrical device, although this process is done via the internet, but it can be done by using a local network too. [1]

R. Piyare et.al presented a paper on Smart Home-Control and Monitoring System using Smart Phone. They proposed a budget friendly and flexible home control and monitoring system using an embedded micro-web server, with IP connectivity for controlling devices and accessing the appliances using an Android based Smart phoneapp.

The mentioned system offers an unconventional communication protocol to observe and control the home environment with more than just the switching functionality. [4]

R.K Kodali et.al proposed a paper on IOT based smart security and home automation system. This IoT project focused on building a smart wireless home security system that sent alerts to the owner by using Internet in case of any unwanted trespasses and raised an alarm. By making the use of same set of sensors, the same can also be utilized for home automation too. [5]

### 3. Components Employed

#### 3.1 Raspberry Pi

The Raspberry Pi is basically a small full-featured computer on a single board that plugs into a computer monitor or TV and uses a standard keyboard and mouse. Due to this, it is capable of doing everything from browsing internet and playing high-definition videos, video games, to making spreadsheets and word processing.

A credit-card sized computer, the Raspberry Pi has the ability to interact with the outside world and provides a great environment for learning programming and digital making.

The Raspberry Pi can also be built into custom projects such as interactive museum exhibits or home automation solutions.

Raspberry Pi 3 is the third generation of raspberry Pi and the purpose of using it in the project is that it has a 64-bit quad core processor, on board Wi-Fi, Bluetooth and USB capabilities.[6] This generation has been described to be ten times faster than the performance of Raspberry Pi 1. A powerful feature of Raspberry Pi is the row of 40 GPIO (general purpose input/output) pins along the top edge of the board. Any of these can be designated as input or output pins and used for wide range of purposes.

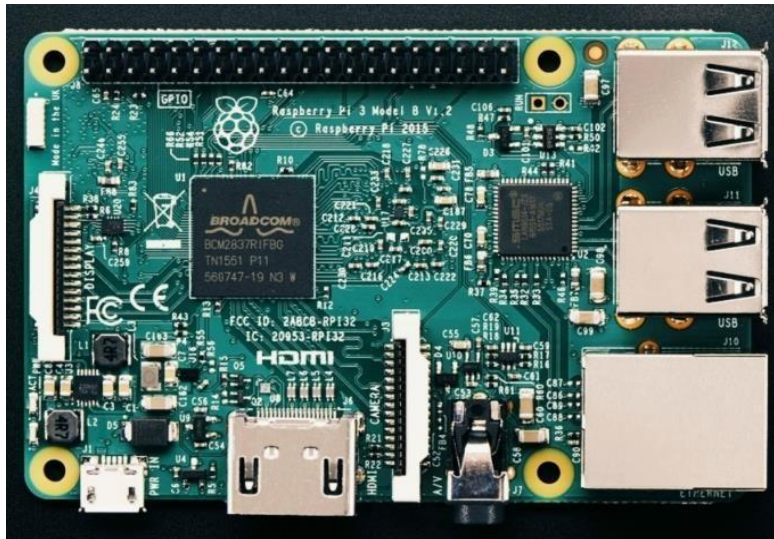


Fig. 1. Raspberry Pi 3

#### 3.2 RelayModule

A relay is an electrically operated switch which means that it can be turned on or off, letting the current go through or not. Relay consists of an electromagnet and a set of contacts. The switching mechanism is carried out with the help of the electromagnet. The input side is a coil while the output side is a switch magnetically connected to the coil. When current flows through the coil, the switch toggles. The electromagnet is activated with a low voltage, for example, 5 volts from a microcontroller and it pulls a contact to make or break a high voltage circuit. [7]

The Relay module is connected to or controlled from the GPIO pins of the Raspberry Pi. It operates on 5V and generally has a threshold of 220V.

Such switches can be extremely useful for a variety of Raspberry Pi projects.

The relay module has 3 pins which are connected to the GPIO pins which are Vcc, that provides the ac power supply to the magnetic coil and controls the opening and closing of a circuit, Ground and Input.

The input pin controls the relay state. When the input 0 is received, circuit is closed. If the input is 1, the circuit is opened. Hence, making it level triggered.

In context to the project, the GPIO pins 1, 6 and 7 are used to connect the relay module to the raspberry Pi board. The Vcc pin of the relay is connected to the pin no. 1 of GPIO which supply the 5V. Ground pin is connected to the pin no. 6 and Input pin is connected to pin no. 7 which is set by the software as an output pin.

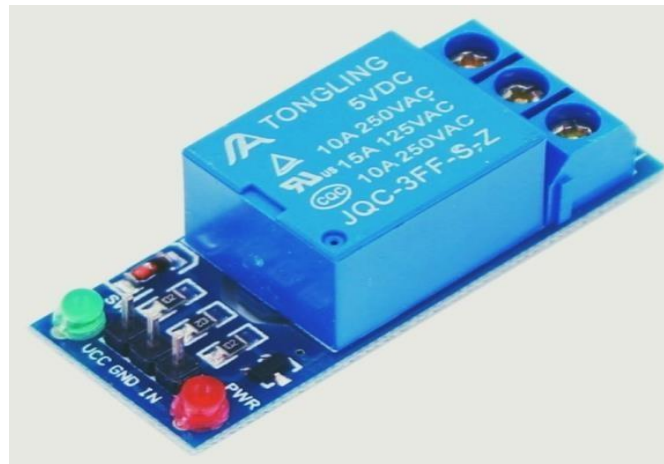


Fig. 2. Relay module

### 3.3 Node.js

Node.js is an open source, cross platform, Javascript run-time environment that executes Javascript code outside of a browser. Originally, the developers of Javascript extended it from something that could only run on the browser to that could run on a machine as a standalone application.

JavaScript is used for client-side scripting and the script is written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser.[8]

Both the JavaScript and Node.js browsers run on the V8 JavaScript runtime engine. This engine takes the JavaScript code and converts it further into a faster machine code. As Machine code is a low-level code which the computer can run without interpreting it first.

Nowadays, JavaScript has the capability to do things that other scripting languages like Python can do. Also, Node.js is superfast and highly scalable which make it great for prototyping and quick development.

### 3.4 TOR

TOR is an acronym that stands for The Onion Router and was initially a worldwide of servers developed with the U.S Navy that enabled people to browse the internet anonymously. However today, it is a non-profit organization whose main objective is to research and development of online privacy tools.

The TOR network works in such a way that it disguises one's identity by moving the traffic across different Tor servers and encrypting that traffic so it is not traced back to the user. If any outside user tries, would see the traffic coming from the random nodes on the Tor network, rather than the user's computer. TOR consists of three-layer proxy. [9] Randomly, Tor browser connects to any one of the publicly listed entry nodes, bounces that traffic through another randomly selected middle relay and then finally spits out the user's traffic through the third and final and exit node. Using TOR makes it difficult to trace any kind of internet activity from visiting to websites, online posts to any communication forms.

Further, it is extended to protect the user's personal privacy as well as their freedom and ability to conduct confidential communication by keeping their internet activities from being monitored.

## 4. Issue to be addressed

A smart home sector is a place where the internet of things meets the household department.

Home automation system doesn't just mean luxury, but it will also provide more efficiency in one's household and also provide a much more convenient and comfortable way of living. As mentioned earlier the smart switch is the basic building block and the initial step of making a home automation system.

The project is web based smart switch using Raspberry Pi 3, controlling any electrical appliance at home from a web browser or a Smartphone is possible with the connectivity capabilities of the Raspberry Pi 3 development board. As this smart switch is capable of being used from anywhere of the world unless and until there is an internet connection. So this internet connectivity has to be strong enough to be operate the switch.

While performing a run-test, it was concluded that the system performed well when connected to 4G connection in comparison to 3G. Hence, a lower latency increases the efficiency of the system.

Relay modules are attached to the board through GPIO pins. Relays are controlled through a JavaScript program on an IP address port. Accessing this IP address port from anywhere in the world is achieved by creating a TOR server. Extending any project which interacts directly with the home network and its appliances, exposes many security risks. This project bypasses this issue by using a TOR based connection. Tor is not meant to completely solve the issue of

**Organized By:** Ansal University, Gurgaon, Haryana.

anonymity on the web but instead it reduces the likelihood for sites to trace the actions and data back to the user. Further, when this smart switch will extend to home security surveillance, TOR will add up to a layer of strong security.

## 5. Flow of Information

### 5.1 FlowDiagram

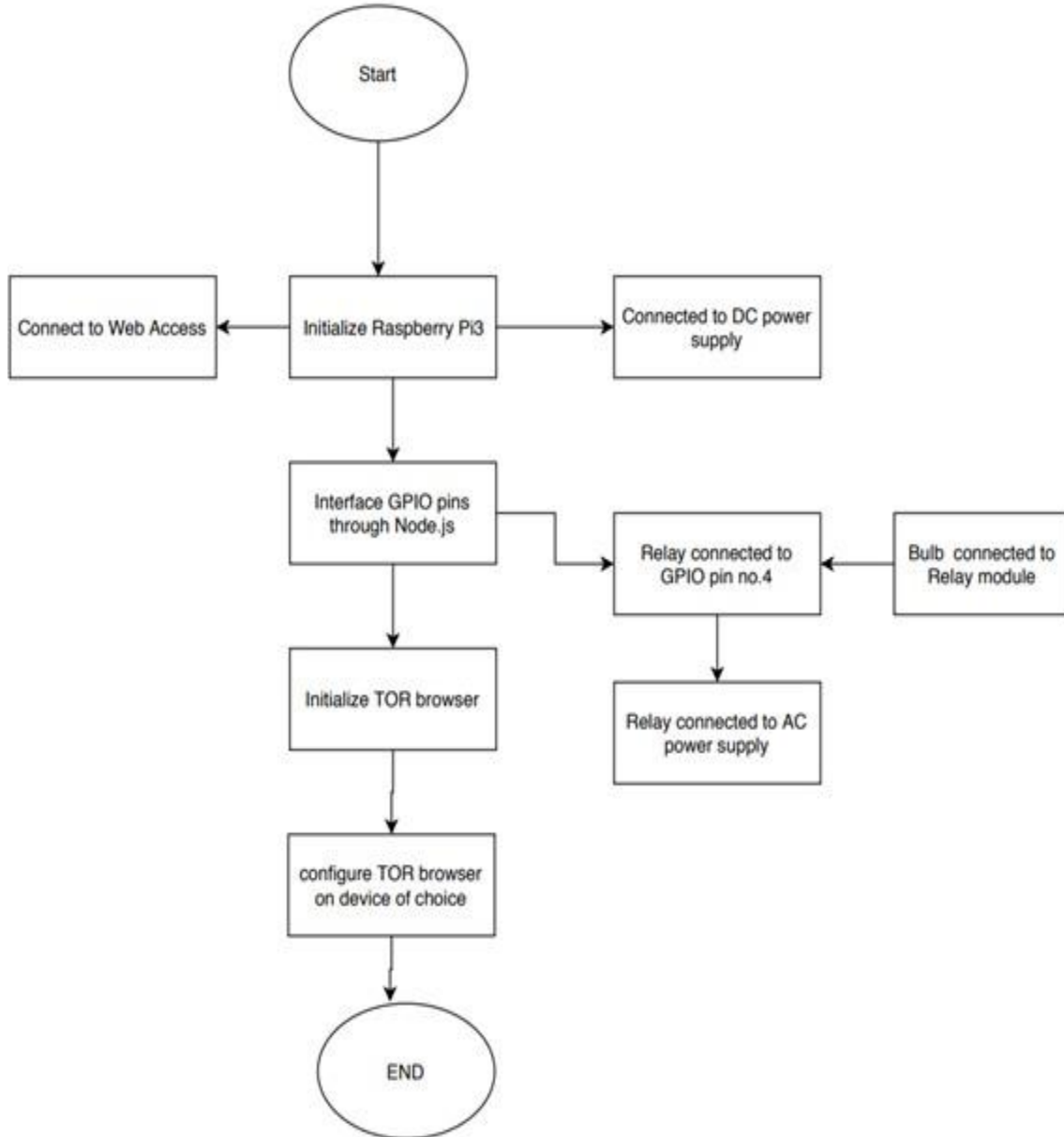


Fig. 3 Flow Diagram of proposed model

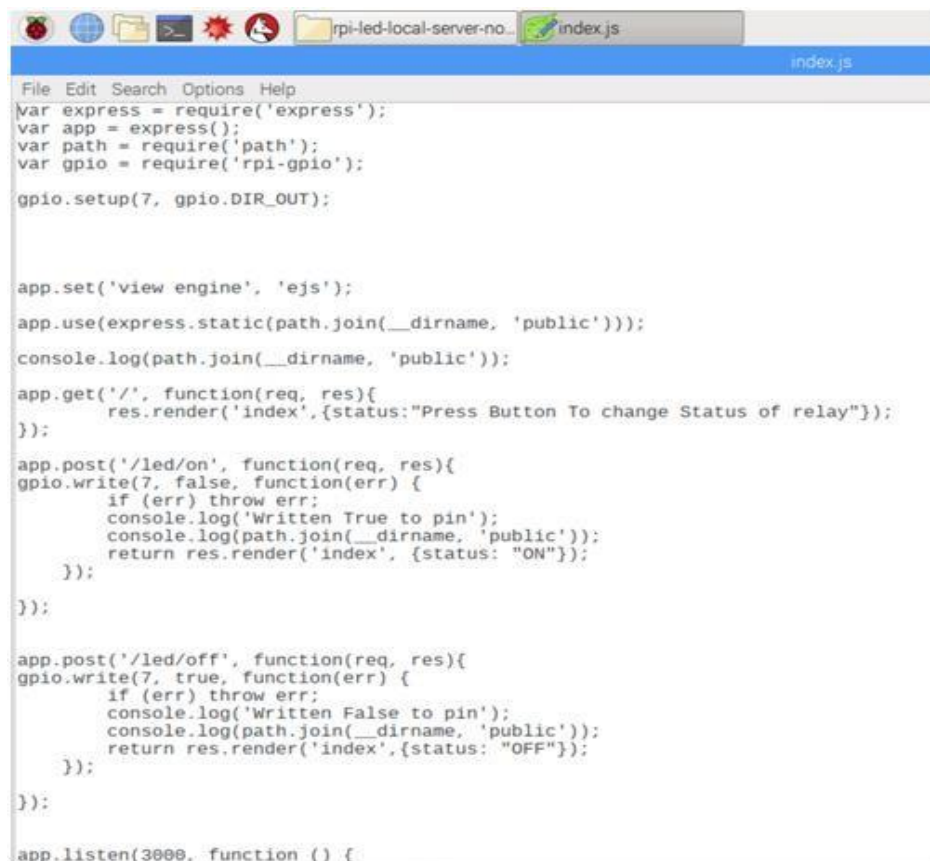
### 5.2 Programcode

#### Interfacing with the gpio pins

After creating a directory named IoT in the root folder, create a new file named index.js

```
gpio.setup(7, gpio.DIR_OUT);
```

```
app.set('view engine', 'ejs');
app.post('/relay/on', function(req, res){
  gpio.write(7, false, function(err) {
    if (err) throw err; console.log('Written
      True to pin');
      console.log(path.join(dirname, 'public')); return
      res.render('index', {status: "ON"});
    });
  });
app.post('/relay/off', function(req, res){
  gpio.write(7, true, function(err) {
    if (err) throw err; console.log('Written
      False to pin');
      console.log(path.join(dirname, 'public')); return
      res.render('index', {status: "OFF"});
    });
  });
});
```



```
File Edit Search Options Help
var express = require('express');
var app = express();
var path = require('path');
var gpio = require('rpi-gpio');

gpio.setup(7, gpio.DIR_OUT);

app.set('view engine', 'ejs');
app.use(express.static(path.join(__dirname, 'public')));
console.log(path.join(__dirname, 'public'));
app.get('/', function(req, res){
  res.render('index', {status: "Press Button To change Status of relay"});
});
app.post('/led/on', function(req, res){
  gpio.write(7, false, function(err) {
    if (err) throw err;
    console.log('Written True to pin');
    console.log(path.join(__dirname, 'public'));
    return res.render('index', {status: "ON"});
  });
});
app.post('/led/off', function(req, res){
  gpio.write(7, true, function(err) {
    if (err) throw err;
    console.log('Written False to pin');
    console.log(path.join(__dirname, 'public'));
    return res.render('index', {status: "OFF"});
  });
});
app.listen(3000, function () {
```

Fig. 4 Interfacing of pins

### Web interface for controlling GPIO

Create a new directory inside the IoT directory named views Make a new file named Index.ejs

Insert the following code in the file

```
<form action="/relay/on" method="post">
  <button type="submit" class="button">On</button>
  <button type="submit" formmethod="post" formaction="/relay/off"
class="buttonbutton3">Off</button>
</form>

<a>Relay Status: <%=status %></a>
</div>

<!-- The below code is only for look and styling -->
<style>
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white; padding:
  15px 32px; text-align:
  center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px; margin:
  4px 2px; cursor: pointer;
}

.button2 {background-color: #008CBA;} /* Blue */
.button3 {background-color: #f44336;} /* Red */
.button4 {background-color: #e7e7e7; color: black;} /* Gray */
.button5 {background-color: #555555;} /* Black */
}
</style>
```

### TORInitialization

```
HiddenServiceDir /var/lib/tor/iot/
HiddenServicePort 80 ip
address:3000HiddenServiceAuthorizeClient stealth
iotremote
```



```
File Edit Search Options Help
<meta name="viewport" content="width=500, initial-scale=1"> <!--This line of code is for mobile responsive -->

<div class="BorderMargin">

  <form action="/led/on" method="post">
    <button type="submit" class="button">On</button>
    <button type="submit" formmethod="post" formaction="/led/off" class="button button3">Off</button>
  </form>

  <a>Relay Status: <%=status %></a>
</div>

<!-- The below code is only for look and styling -->
<style>
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}

.button2 {background-color: #009688;} /* Blue */
.button3 {background-color: #f44336;} /* Red */
.button4 {background-color: #9e9e9e; color: black;} /* Gray */
.button5 {background-color: #555555;} /* Black */

.BorderMargin {
  margin: 10px;
  padding: 10px;
  max-width: 200px;
  height: 100px;
  border: 1px solid black;
}
</style>
```

Fig.5 Initialization of TOR

**Starting the program**

Open a web browser on the same network and type in  
<http://IPAddress:3000/>

Pressing on will turn on the relay and conversely pressing off will turn off the relay

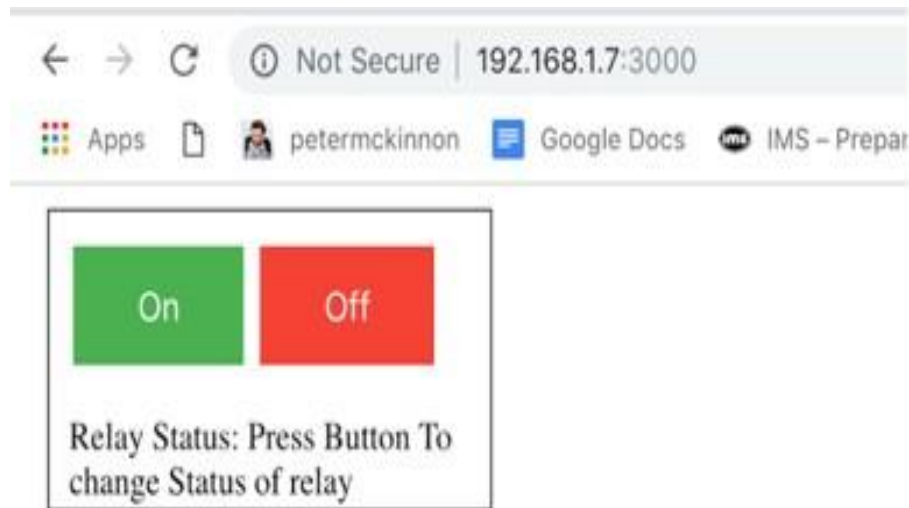


Fig.6 Web page to be accessed

For accessing the port from anywhere in the world Run the “dot onion” site obtained for TOR in a TOR enabled browser

### 5.3 Workingmodel

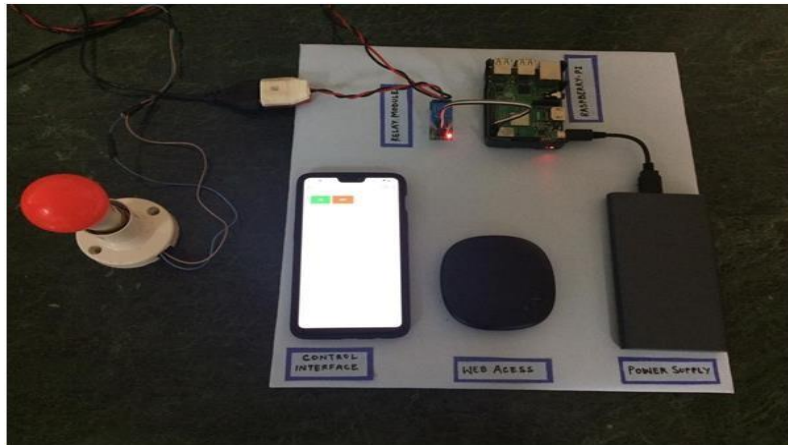


Fig. 7 When the switch is OFF

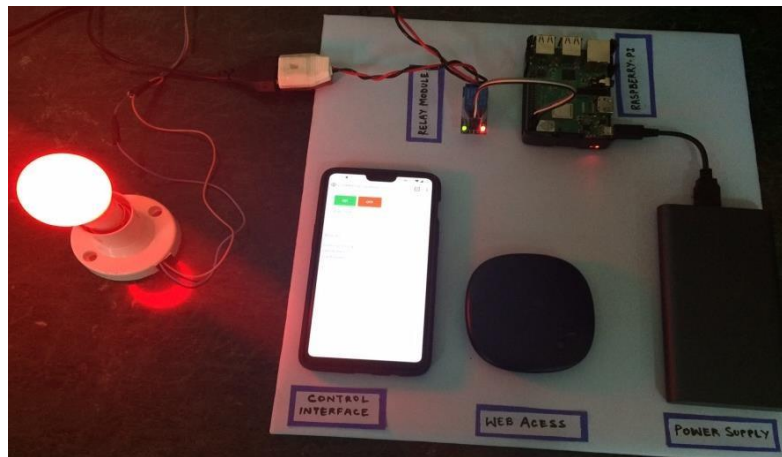


Fig. 8 When the switch is ON

## 6. Conclusion

After implementing this project, the user will be able to control any electrical appliance of their choosing. By applying various technologies working in conjunctions, a safe connection is made through the internet for enabling home automation. Raspberry Pi provided a small, affordable and simple to use board for implementing this project.

[10]It also provided on-board connectivity chips, reducing the need to buy more separate components hence making it more efficient. Relay module enabled this project to control any physical appliance with a simple binary command.

Node.js was used as a platform to build a small footprint website which enabled the user to interface GPIO pins of the board. A small size will ensure faster response and loading time. [11]

TOR server was used to provide remote access to the website, bypassing the need to build a server. TOR also provided an iron clad security to the website by anonymizing the pathway for gaining access to the GPIO pins.

## 7. Futurescope

This project can be further integrated with a home security system and act as a user initiated alarm. As the system is dependent on the user's circumspection and discernment of the situation (whether someone known, or an intruder is entering the house), the use of a camera connected to the Raspberry Pi might help the user in taking decisions whether to activate the alarm system or welcome the guest. The snapshot of the guest or intruder can be taken and mailed to the user. The user can further forward the same photograph to the police station if he or she wishes. [12] Any home security is rendered useless until



it can be accessed outside a local network. With the advent of 5G networks, the latency of turning on and off will be reduced significantly.

This project will enable a user to control lights or alarm system with the addition of some motion sensors.

To know, whether or not an area is being trespassed by an intruder, a live stream of the area can also be recorded.

## **8. References**

- [1] Roberto Morales- Caporal, J.J Rodriguez-Rivas, J.E Giral-Salas, Edmundo Bonilla Huerta: A Smart Switch to connect and disconnect Electrical Devices at home using Internet: IEEE Latin America Transaction(2016), Vol 14(4),pp.1575-1581.
- [2] Home Automation Wikipedia link:[https://en.wikipedia.org/wiki/Home\\_automation](https://en.wikipedia.org/wiki/Home_automation)
- [3] Home Automation History link:<https://www.somfy.com.au/Blog>
- [4] Rajeev Piyare, Seong Ro Lee: Smart Home- Control and Monitoring System Using Smart Phone: 1st International Conference on Convergence and its application(ICC),Volume24,pp.83-86
- [5] Ravi Kishore Kodali, Vishal Jain, Suvadeep Bose, Lakshmi Boopana: IoT based smart security, and home automation system: 2016 International Conference On Computing Communication And Automation(ICC), (April 2016), pp.1286-1289
- [6] Raspberry Pi Wikipedia link:[https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- [7] Relay module link:<https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/>
- [8] Node.js link:[https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)
- [9] TOR server link:<https://www.torproject.org/projects/torbrowser.html.en>
- [10] Raspberry Pi link: [https://www.w3schools.com/nodejs/nodejs\\_raspberrypi.asp](https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp)
- [11] Interfacing with Node.js link:<https://nodejs.org/api/addons.html>
- [12] Davinder Sharma, Avatar Baldeo, Cassiel Phillip: Raspberry Pi based Smart Home for deployment in the Smart Grid, International Journal of Computer Applications(0975-8887), Vol. 119-No.4,June2015,pp.6-10