Lab 08 - Web control of IOT using Apache server in Raspberry Pi/ Arduino

Name of student (Batch No / Roll No)

Divyang Bagla (D2/PD33)

Performance of Experiment	Journal Submission	Total Marks	Remarks	Instructor Sign

Aim: To install web server such as Apache Web Server in IoT platform and write web application to access the server (IoT device as Web Server)

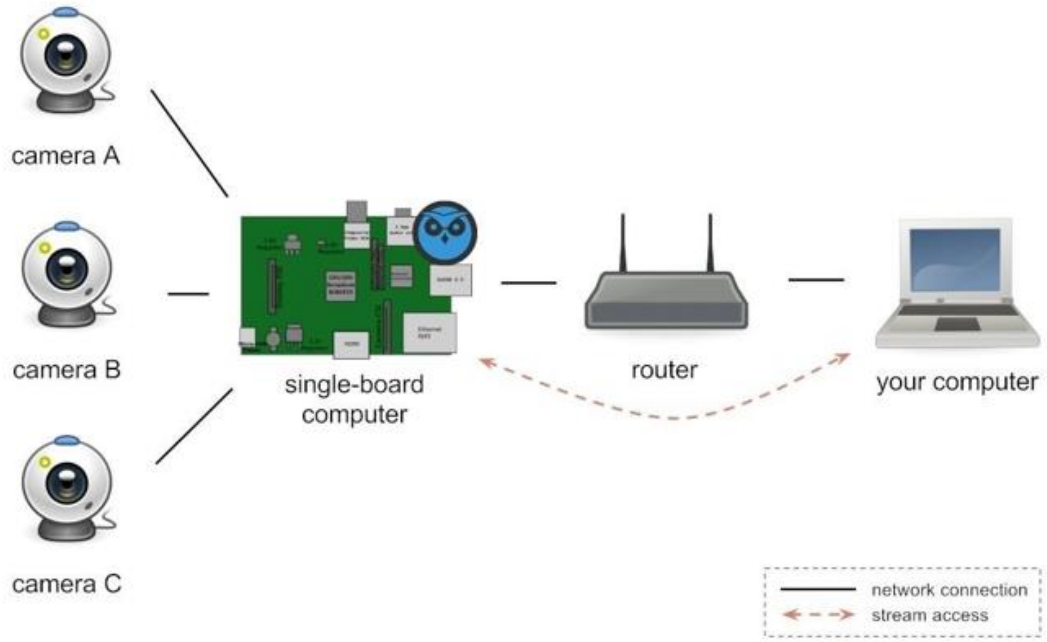
Journal content- Theory and frequently asked questions and experiment
Objectives:

1. To understand web and cloud technologies use to empower IoT applications
2. To demonstrate web formatted communication or web interface with Apache server which is running on Raspberry pi3. These web request control LEDs but in reality it can be any sensor or actuator.
3. To understand html, python and php web programming to carry out web request and control of sensor or actuator connected with raspberry pi.

Theory and FAQ

1 Draw system diagram showing different hardware and software components to show raspberry pi as web server.
2. State few applications which needs web system control over sensors or actuators.
3. Sketch layout of system
4. Screen out of web program to put LED on or OFF
5. State briefly other web servers which are better for raspberry pi as web server.
6. State any other programming which can have web control similar to PHP
7. List and state online IoT platforms that eases work IoT application development.
8. What is IP, DNS?
9. List and state Amazon IoT platform services
10. List and state IBM IoT platform services

1. Draw system diagram showing different hardware and software components to show raspberry pi as web server.



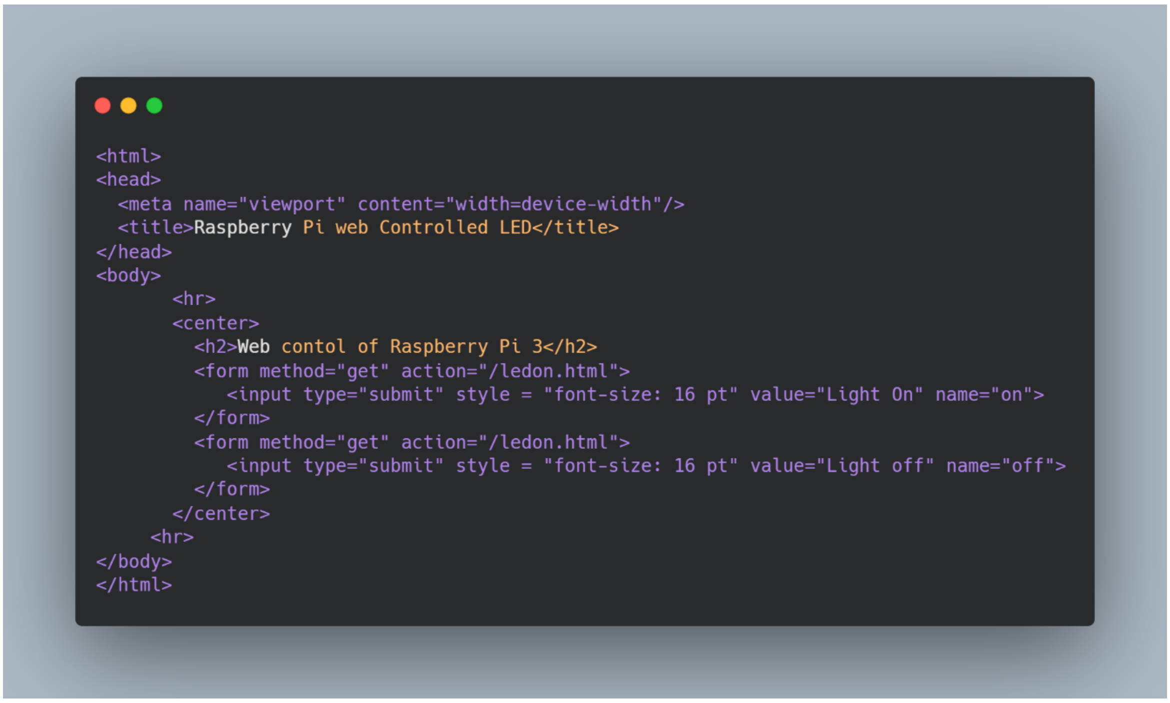
A **web server** can, in general, contain one or more websites. ... The primary **function** of a **web server** is to store, process and deliver **web** pages to clients. The communication between client and **server** takes place using the Hypertext Transfer Protocol (**HTTP**).

For many Internet of Things (IoT) applications, embedding a HTTP/WebSocket server on a device is enough to get the job done. Oftentimes, this solution can be faster, simpler and more cost effective. Choosing the right web server can be difficult.

2. State few applications which needs web system control over sensors or actuators.

- Using web application the system is easier to use world wide.
- There is a system for monitoring the temperature that can be accessible from anywhere, and store the received information. The developed system is based on Raspberry PI and sensor node. The node is composed of two different temperature sensors. The average value from the sensors is stored in database, and later will be published on the web.

3. Screen out of web program to put LED on or OFF



```html
<html>
<head>
  <meta name="viewport" content="width=device-width"/>
  <title>Raspberry Pi web Controlled LED</title>
</head>
<body>
       <hr>
       <center>
         <h2>Web contol of Raspberry Pi 3</h2>
         <form method="get" action="/ledon.html">
            <input type="submit" style = "font-size: 16 pt" value="Light On" name="on">
         </form>
         <form method="get" action="/ledon.html">
            <input type="submit" style = "font-size: 16 pt" value="Light off" name="off">
         </form>
       </center>
     <hr>
</body>
</html>
```

4. State briefly other web servers which are better for raspberry pi as web server.

1. Apache HTTP Server

The Apache HTTP Server—often referred to as httpd, or simply Apache—was first launched in 1995, and celebrated its 20th birthday in February 2015. Apache powers 52% of all websites globally, and is by far the most popular web server.

While Apache httpd is most often seen running on Linux, you can also deploy Apache on OS X and Windows. Apache is, unsurprisingly, licensed under the Apache License version 2. The web server itself uses a modular architecture, in which extra modules can be loaded to extend its features. For example, loading the mod_proxy will allow for a proxy/gateway on your server, and mod_proxy_balancer will enable load balancing for all supported protocols. As of version 2.4, Apache also supports HTTP/2 through a new module, mod_http2.

2. NGINX

Igor Sysoev began developing NGINX back in 2002, with its first public release in 2004. NGINX was developed as an answer to the so-called C10K problem, which is shorthand for "how do you design a web server which can handle ten thousand concurrent connections?" NGINX is second on a list of open source web servers by usage, running just over 30% of all websites.

NGINX relies on an asynchronous event-driven architecture to help power its goal of handling massive concurrent sessions. It has become a very popular web server among administrators due to its light resource utilization and its ability to scale easily.

3. Apache Tomcat

Apache Tomcat is an open source Java servlet container that functions as a web server. A Java servlet is a Java program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted on Web servers. Such web servlets are the Java counterpart to other dynamic web content technologies such as PHP and ASP.NET. Tomcat's code base was donated by Sun Microsystems to the Apache Software Foundation in 1999, and became a top-level Apache project in 2005. It currently powers just under 1% of all websites.

Apache Tomcat, released under the Apache License version 2, is typically used to run Java applications. It can, however, be extended with Coyote, to also perform the role of a normal web server that serves local files as HTTP documents. More information can be found on the project website.

4. Node.js

Node.js is a server-side JavaScript environment for network applications such as web servers. With a smaller market position, Node.js powers 0.2% of all websites. Node.js was originally written in 2009 by Ryan Dahl. The Node.js project, governed by the Node.js Foundation, is facilitated by the Linux Foundation's Collaborative Projects program.

The difference between Node.js and other popular web servers is that it is primarily a cross-platform runtime environment to build network applications with. Node.js applies an event-driven architecture capable of asynchronous I/O. These design choices optimize throughput and scalability in web applications allowing to run real-time communication and browser games. Node.js also highlights the difference in web development stacks, where Node.js is clearly part of the HTML, CSS, and JavaScript stack, as opposed to Apache or NGINX which are a part of many different software stacks.

5. State any other programming which can have web control similar to PHP

- *JavaScript*
- *Java*
- *Python*
- *Ruby*
- *C#*
- *Go*

6. List and state online IoT platforms that eases work IoT application development.

Azure IoT Suite

A collection of services as such IoT Hub, Stream Analytics, Machine Learning, Notification Hubs, and Power BI make up the Azure IoT Suite. Powering a plethora of devices and operating systems, it allows analysis and visualization of more data. Hence, it is also recognized for its feature like scalability from POC to the larger deployment stage.

Let's have a look at what the enterprise-grade services offer:

- Data collection from devices
- Analyze data streams in-motion
- Store and query large data sets
- Visualize real-time as well as historical data
- Back-office systems integration
- Device management

IBM Watson

With **IBM Watson**, you can quickly secure all types of connected devices. It allows developers to have access to complete connectivity, information management, and real-time data analysis along with risk and security management.

The real-time data collected from the connected devices are managed, then configured and integrated with data services. Securely designed, this platform has given developers an excellent way to protect the integrity of your IoT solution.

Amazon Web Services (AWS)

AWS is one of the most popular IoT platforms on the market and offers a highly reliable and low-cost infrastructure platform in the cloud. Due to its great scalability, it is used by thousands of businesses all around the world, and its cost-effectiveness and flexibility make this platform one of the best option for developers.

Oracle IoT

Oracle hit the IoT market like no other platform. Released in late 2016, it holds a crucial place. With the cloud service, you can receive flexible **IoT network topologies**, simulations of a virtual device, stream processing, and command and control functionality.

Furthermore, it allows users to collect data securely from any device in any market through technical abstraction. Oracle IoT is pre-integrated with Oracle PaaS and on-premise applications using open APIs. Hence, it provides real-time analytics via big data processing.

Kaa

Kaa is a 100% open source IoT platform that provides a unique development experience for end-to-end IoT solutions, smart products, and connected applications. Due to its feature-rich toolkit, this IoT platform fits your budget and, at the same time, is customizable for different business applications. Moreover, it offers easy and direct hardware integration to developers.

7. What is IP, DNS?

IP Address

An IP address is a numerical representation of where a device is connected to the internet. It’s how you identify where something is and, to some degree, what that thing is. Understanding the basics of IP addresses is essential to making your way around the internet.

DNS

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

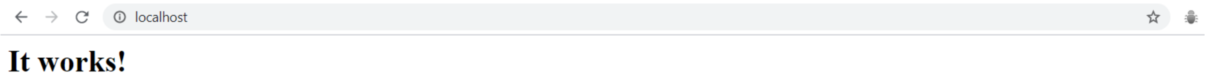
8. List and state Amazon IoT platform services

- Device software. Connect your devices and operate them at the edge.
- Connectivity & control services. Secure, control, and manage your devices from the cloud. AWS IoT Core. ...
- Analytics services. Work with IoT data faster to extract value from your IoT data. AWS IoT Analytics.

9. List and state IBM IoT platform services

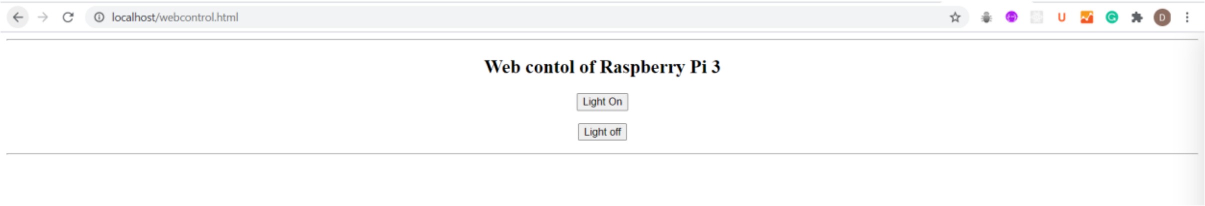
- From chip to app to cloud, take full advantage of cognitive Watson APIs, visual dashboards, rich developer resources and industry-leading security to accelerate enterprise IoT insight.
- Identify, aggregate and transform data from your IoT sources into asset-based data structures.
- Manage risk and gather insights across your entire IoT landscape, using dashboards and sophisticated alerts to monitor devices, apps, and connections.
- Quickly and easily parse, filter, and transform device and performance data. Cache or archive data selectively for off-platform analytics or for integration with IoT apps.

SERVER RUNNING :



It works!

Loaded Webcontrol page :



Web contol of Raspberry Pi 3

Conclusion

Thus we could control sensors and actuators using web interface. Since web interaction has many advantages over traditional interfacing and control of IoT devices, it is advisable to prefer web control over IoT devices. However we need to take care of security, sessions and stateless nature of Internet Technology.