

# Implementation of Linux Web Server for Web-Based Automation System Using Raspberry Pi

Amit S. Wankhade<sup>1</sup>, Daithala Sreedhar<sup>2</sup>, Dr. B. Satyanaryana<sup>3</sup>

Student, M. Tech., Software Engg, Department of CSE, VBIT, Janagaon, Warangal, TS, India (JNTU)<sup>1</sup>

HOD, Department of CSE, VBIT, Janagaon, Warangal, TS, India (JNTU)<sup>2</sup>

Professor of CSE & Principal, VBIT, Janagaon, Warangal, TS, India (JNTU)<sup>3</sup>

**Abstract:** The Raspberry Pi changed the way many engineers and developers thought about the possibilities of an embedded system. When networking technology is combined with it, the scope of embedded systems would be further more. Here, this paper focuses on implementation of Linux web server and web-based automation system. And also give information about how to configure the apache web server, MySQL for database support, and PHP for dynamic web content. The sample web page demonstrates automation to control the raspberry end connected devices via relay board. The primary idea is to make efficient and effective web based automation system at very low cost by using the Raspberry Pi 3 B model.

**Keywords:** Web Server; MySQL; HTML; PHP; SSH.

## I. INTRODUCTION

The Internet is always available for you to use, you don't have to wait for it to open or wait for someone to make it available. So when the embedded devices are provided with internet access, it is of no doubt that demand will rise due to the remote accessing capability of the devices. The paper includes complete implementation of an HTTP Web Server in Raspberry Pi. This development kit which contains Ethernet interface is connected to PC using RJ45 cable. The Raspberry Pi 3 B model is used to implementation of web server and automation. The web site is design in simple HTML which give a good user interface to control the devices which is connected with the Raspberry Pi. The relay board activate as the signal given through web page and perform the ON and OFF operation at real time. In this implementation Raspberry Pi 3 B model is used which is latest launch by company.

### A. The Internet of Things

An ongoing trend in embedded devices is to have all embedded devices connected to the Internet. The Internet was developed as a failsafe network that could survive the destruction of several nodes. The Internet of Things (IoT) leverages the same redundancy. With the move to migrate to IPv6, the IP address space would be large enough for several trillion devices to stay connected. A connected device also makes it very convenient to control it from anywhere, receive inputs from various sensors and respond to events. A multitude of IoT connected devices in a home has the potential to act as a living entity that exhibits response to stimuli.

### B. Raspberry Pi 3 B Specifications

SoC: Broadcom BCM2837

CPU: 4× ARM Cortex-A53, 1.2GHz

GPU: Broadcom Video Core IV

RAM: 1GB LPDDR2 (900 MHz)

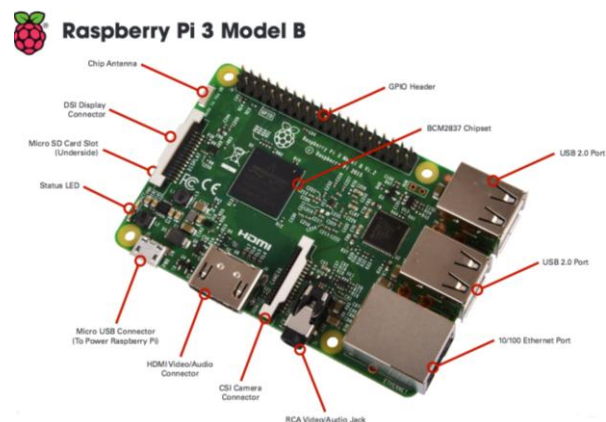
Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless

Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy

Storage: micro SD

GPIO: 40-pin header, populated

Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI).



## II. PROBLEM DEFINATION

A Linux web server should use the HTTP protocol to transmit Web pages from the embedded system to the web browser and to transmit form data back to the embedded system attached to the appliance. The embedded system requires a network interface, such as Ethernet, a TCP/IP protocol stack, embedded web server software and static and dynamic web pages that form the user interface for

that specific device. Because the embedded systems have limited CPU and memory resources and these resources are mostly used by real-time applications, end- users may have to wait up to few seconds for an HTTP response. Multi-threading should be employed in the embedded systems to avoid slow response.

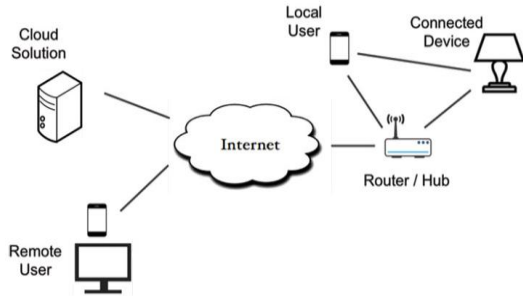


Fig1- System architecture of the proposed Linux web server based automation.

### III. SYSTEM DESIGN

In this paper, an experimental setup is implemented with real instruments. An arduino controller is used along with the raspberry pi to achieve the desired result. A Web-based system is developed using the Smart Access feature of the Raspberry pi. The equipment used in experimental setup is shown in following fig.

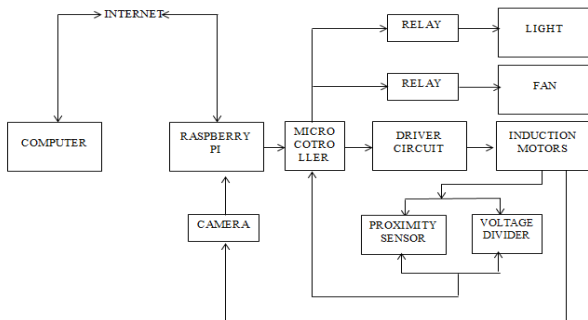


Fig.1 the components of the experimental setup are

### IV. SETTING UP THE WEB SERVER CONFIGURATION

#### A. Initial Setup

##### 1. Apache Web Server

First we need to download the apache web server package, to do this log into your Raspberry Pi via SSH and enter the following command

```
sudo apt-get install apache2
```

This is just the default index.html, in order to edit the page we need to adjust the permissions on the file. Log into your Raspberry Pi and access the webserver folder using the following commands

```
cd /var/www/html
ls al
```

```
pi@raspberrypi: /var/www/html
login as: pi
pi@192.168.1.20's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jun  6 10:42:20 2016 from dominic-pc
pi@raspberrypi:~$ cd /var/www/html
pi@raspberrypi:/var/www/html$ ls -al
total 20
drwxr-xr-x 2 root root 4096 Jun  6 10:43 .
drwxr-xr-x 3 root root 4096 Jun  6 10:43 ..
-rw-r--r-- 1 root root 11104 Jun  6 10:43 index.html
pi@raspberrypi:/var/www/html$ sudo chown pi: index.html
pi@raspberrypi:/var/www/html$ ls -al
total 20
drwxr-xr-x 2 root root 4096 Jun  6 10:43 .
drwxr-xr-x 3 root root 4096 Jun  6 10:43 ..
-rw-r--r-- 1 pi pi 11104 Jun  6 10:43 index.html
pi@raspberrypi:/var/www/html$ nano index.html
```

This shows you the index.html file that was presented to your browser, you will notice that its permissions are currently set to the root user. We need to change the user permissions if we want to make any changes to the file.

```
sudo chown pi: index.html
nano index.html
```

nano command is used to edit the index file, so Here you will see the HTML code for you index page, if you scroll down to the <body> section of the page you will see the writing that was presented in the browser, change some of this then press Ctrl+X, and Y to save the changes. If you now go back to your browser and refresh you should see your changes.

```
pi@raspberrypi:~$ cd /var/www/html
pi@raspberrypi:/var/www/html$ ls -al
total 20
drwxr-xr-x 2 root root 4096 Jun  6 10:43 .
drwxr-xr-x 3 root root 4096 Jun  6 10:43 ..
-rw-r--r-- 1 root root 11104 Jun  6 10:43 index.html
pi@raspberrypi:/var/www/html$ sudo chown pi: index.html
pi@raspberrypi:/var/www/html$ ls -al
total 20
drwxr-xr-x 2 root root 4096 Jun  6 10:43 .
drwxr-xr-x 3 root root 4096 Jun  6 10:43 ..
-rw-r--r-- 1 pi pi 11104 Jun  6 10:43 index.html
pi@raspberrypi:/var/www/html$ nano index.html
GNU nano 2.2.6 File: index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/html4/01/transition.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8" />
<title>Apache2 Debian Default Page: It works/alive!
</title>
</head>
<body>
<div style="text-align:center">
<h1>It works!</h1>
</div>
</body>
</html>
```

##### 2. Install PHP

To download the PHP package log into your Raspberry Pi via SSH and enter the following command

```
sudo apt-get install php5 libapache2-mod-php5 -y
```

The benefit of PHP webpage scripting is that it presents dynamic pages based on changing information, if you want to see this in action try changing your index.php file with the following entry

```
<?php echo date('Y-m-d H:i:s'); ?>
```



### 3. The MySQL Database

MySQL provides us with a database to store any information we require for our website, it is used extensively with many websites on the internet today. To add this capability to our Raspberry Pi we again need to load a couple of packages.

```
sudo apt-get install mysql-server php5-mysql -y
```

### 4. Connection to a Domain Name Server (DNS)

DNS is like the internet's address book, allowing others to find your website by looking up your domain name. I use Namecheap as they can provide both of these services while you do the hosting yourself on the Pi.

A domain name generally costs around Rs. 250 to 700 a year depending on the name you choose, Namecheap also provide free DNS. Just follow the link below to their website, find an available domain name and you should be up and running in no time.

### B. Server Configuration

To begin the process of configuring a Raspberry Pi to work as web server, an Operating System of a Linux Distribution, called RASBIAN, specifically created for the Raspberry Pi, was downloaded onto an SD memory card with at least 4 GB of memory, via another computer, and inserted into the Raspberry Pi. Details of the completed installation can be viewed by opening the website and logging in with the correct user credentials. A Windows Distribution or a Mac OS X can be installed instead, Linux was a personal preference.

And now RASBIAN is the official operating system for Raspberry Pi. There are many different methods to finish the configuration and installations of your Raspberry Pi to make it work as a web server, we will stick to the methods we used to accomplish this, if you are interested in other methods they are easily found all over the internet.

The rest can be accomplished through following simple instructions or tutorials online. Since we enabled SSH we were then able to use to finish the setup and installations from a laptop. We used the CMD console to access the pi with SSH and used SUDO and Linux bash commands.

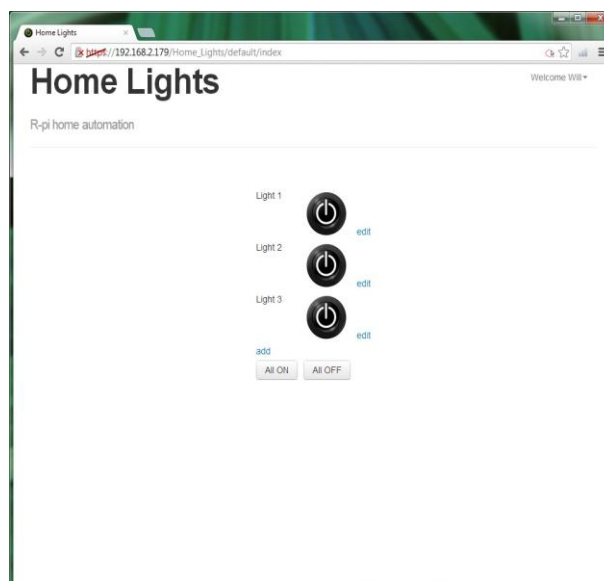
As mentioned earlier, we used port forwarding to access the Raspberry Pi from another source; another computer, phone, etc. To do this, the source would have to attempt to gain access through the router where the Raspberry Pi is connected.

This can be accomplished by typing in the Public IP address 192.168.2.179 and either port 80, 21, 22, or 23 into the URL bar as so: 192.168.2.179:80. This will forward any traffic attempting to access the Public IP address towards the Local/Private IP address 192.168.1.6 of Raspberry Pi, thus granting us remote access to the Raspberry Pi and its contents

```
pi@HackerPi:~$ sudo apt-get install apache2
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module status.
Enabling module auth_basic.
Enabling module deflate.
Enabling module authz_default.
Enabling module authz_user.
Enabling module authz_groupfile.
Enabling module authn_file.
Enabling module authz_host.
Enabling module reqtimeout.
Setting up apache2-mpm-worker (2.2.22-13+deb7u6) ...
[...] Starting web server: apache2apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
. ok
Setting up apache2 (2.2.22-13+deb7u6) ...
Setting up ssl-cert (1.0.32+deb7u1) ...
pi@HackerPi ~$ sudo service apache2 restart
[...] Restarting web server: apache2apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
... waiting .apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
. ok
```

## V. WEB-BASED CONTROL

The design of the website was done using PHP designed. The web page has a authentication page as the homepage.



The control page is used in controlling the home appliances and for viewing the video stream from the camera. The IP address along with date and time of the last visited system will be displayed on the site.

When the required device is selected and the set button is clicked the signal is sent to the controller and the device gets on or off as requested.

## VI. ADVANTAGES

Circuit is portable and could be used anywhere. The size of a raspberry pi is of credit card size. Also automatic controlling of circuit reduce manpower. Cost is less to create an embedded Web server hence rather than buying a real costly web server an embedded web server is a cost efficient solution. It take very less time make modification or changes in parameters and easy to host web page globally.

## VII. CONCLUSION

We were to setup a web server in very small Budget. So we planned to use a Raspberry Pi to configure and set up web server with an IP address and port forwarding, which would allow access from another source connected to a Network. In this case the given web server has minimal features. The main focus is more on the development and configuration process with a very fundamental format that makes it easy to understand and simple to teach to someone who has little to no experience setting up and configuring a web server with a Raspberry Pi. We used very basic Python commands, some Linux fundamental commands, and some HTML & PHP language basics.

## VIII. FUTURE SCOPE

This System can be used as cloud storage. A download and upload link can be provided to the website and this server can also be used as a cloud storage. We can interface more sensors and control processes. We have use to two sensors. Also to make it very simple it can be used just as just a monitoring system also in an Industrial automation this system is very useful to control big devices like motors and other heavy electrical machinery. Also in agriculture this system will very useful in different irrigation and also it will minimize the farmer's efforts.

## REFERENCES

- [1] Web Server for Embedded Systems, Klaus-D. Walter, 2005
- [2] G. -J. Han, M. Guan and H. Zhao. "EWS: Providing Internet Connectivity for non-PC Devices," 2004 IEEE Internet Conference on Networking, Sensing and Control (IEEE ICNSC'04), Taipei, Taiwan, Vol. 1, March 21 - 23, 2004, pp. 349-354.
- [3] Raspbiserver.no-ip.org/phpmyadmin Last visited Feb 2015
- [4] <http://www.scribd.com/doc/255957808/Raspberry-Pi-forBeginners-Revised>. Last visited March 26th 2015
- [5] X. Peng, J. W. Xia and J. M. Liao, "An Embedded Inter-net Interface System," Mobile Robots, Vol. 8, No. 2, 2001, pp. 520-531.
- [6] Patinge S, Suryawanshi Y, Kakde S. Design of ARM based data acquisition and control using GSM and TCP/IP Network. 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCC); 2013.
- [7] M Poongothai. ARM embedded web server based on DAC system. 2011 IEEE International Conference on Process Automation, Control and Computing (PACC); 2011.
- [8] Raspberry pi education manual, Version 1.0 December 2012
- [9] Pritish Sachdeva and Shrutik Katchii, "A Review Paper on Raspberry Pi", Vol.4, No.6, Dec 2014
- [10] NEWS BRIEF, Published by the "IEEE Computer Society", 0018-9162/12 © 2012
- [11] Eben Upton, Gareth Halfacree, "Raspberry Pi User Guide manual" 2012.
- [12] Raspberry Pi Start Guide.RS Components, Vsn 1.0 3/2012
- [13] Trifan, A., Neves, A.J.R. & Cunha, B, Evaluation of color spaces for user-supervised color classification in robotic vision.17th International Conference on Image Processing, Computer Vision, & Pattern Recognition, Las Vegas, Nevada, USA (July 2013).
- [14] R. Cucchiara, C. Grana, M.Piccardi& A. Prati, Detecting moving objects, ghosts, and shadows in video streams, IEEETransactions on Pattern Analysis and Machine Intelligence(PAMI), Vol. 25(10), 1337 - 1342, 2003.pp.25-28
- [15] S. Tuohy, D. O'Cualain, E. Jones, & M. Glavin, Distance determination for an automobile environment using inverse perspective mapping in OpenCV, in Proc. Irish Signals and Systems Conference 2010.
- [16] Li, M., Zhao, C., Hou, Y. & Ren, M. , A New Lane Line Segmentation and Detection Method based on Inverse Perspective Mapping, International Journal of Digital Content Technology and its Applications. Volume 5, Number 4, April 2011, pp. 230-236
- [17] Dhaval Chheda, Divyesh Darde & Shraddha Chitalia, Smart Projectors using Remote Controlled Raspberry Pi, International Journal of Computer Applications (0975 – 8887),Volume 82 – No. 16,2013, pp.6-11
- [18] Stewart Watkiss, Design and build a Raspberry Pi robot [Online], availableat:<http://www.penguintutor.com/electronics/robot/rubyrobot-detailedguide.pdf>
- [19] David Hayward, Raspberry Pi operating systems: 5 reviewed and rated[Online],availableat:<http://www.in.techradar.com/news/software/>
- [20] Matt Richardson, Shawn Wallace, Getting Started with Raspberry Pi, 2nd Edition, Published by Maker Media, Inc., USA, 2014. Book ISBN: 978-1-457-18612-7 [21] Gary Bradski, Adrian Kaehler, and Learning OpenCV: Computer Vision with the OpenCV Library, "O'Reilly Media, Inc.". Copyright.September 2008, 1st Edition, Book ISBN: 978-0-596-51613-0
- [22] Ch. Krishnaveni , Ms. A. Siresha , Mr. B. J. Prem Prasana Kumar & Mr. K. Sivanagireddy, Implementation of embedded systems for pedestrian safety using haar features, IJEC: International Journal of Electrical Electronics and Communication,ISN 2048 – 1068,Volume: 06 Issue: 201 Oct -2014, pp. 761-766
- [23] Tan-Hung Duong, Sun-Tae Chung, Seongwon Cho, Model-Based Robust Lane Detection for Driver Assistance, available at: [http://www.kpubs.org/article/articleMain.kpubs?spotType=low&articleANo=MTMDCW\\_2014\\_v17n6\\_655](http://www.kpubs.org/article/articleMain.kpubs?spotType=low&articleANo=MTMDCW_2014_v17n6_655).