University of Texas at San Antonio, Texas

‘Engineering Programming 2’ Term Project (Fall 17)

Instructor: Dr. Qian Chen

Title: Raspberry Pi Wi-Fi Configuration Through Hotspot Interface using JAVA

Submitted By: Gopal Vishwakarma

Department of Electrical and Computer Engineering

Dde911 @01688966

ECE graduate student

GOPAL VISHWAKARMA
Dde911 @01688966

1. Abstract

This project gives insight of Web development and Wi-Fi/Hotspot interface configuration using Java language on Raspberry Pi board. This project is being developed to give facility of Wi-Fi configuration through any PC, tablet and mobiles. Initially, Raspberry Pi will be in Hotspot mode after power on. So, any mobile/tablet/PC can able to connect to Raspberry pi through this interface. Raspberry Pi hosts a webpage at 192.168.50.1 address. Tomcat server is being deployed on Rpi for hosting webpage. Raspberry Pi lists out available Wi-Fi in the vicinity and java program gets the Wi-Fi list and provides information on webpage. Raspberry Pi connects to router by selecting SSID and entering password for router on webpage. This project gives facility of configuration Wi-Fi interface without need of display monitor and keyboard.

2. Introduction:

Raspberry pi 3 is Debian (Linux) based cost effective microcomputer. Raspberry Pi version 3 has on-board Broadcom Wi-Fi chip. Now-a-days, Raspberry pi is widely used in applications like headless embedded computer, IOT/Wi-Fi/GPRS gateway, Advertising displays etc. But mostly in all scenarios, Configuration of Wi-Fi settings requires technical person with external displays, keyboards, HDMI cables. This project will facilitate user to configure Wi-Fi settings through hotspot interface. So that user can access Raspberry Pi through not only laptop or pc but also through smart mobile. In this project, Raspberry Pi will host a webpage which can be accessed through hotspot interface. User will access webpage through internet browser on his laptop or mobile. Here user will enter Wi-Fi credentials through webpage. So that Raspberry will get connect to Wi-Fi router.

Development languages used: Java, Python, Html, Bash Script

Equipments used:

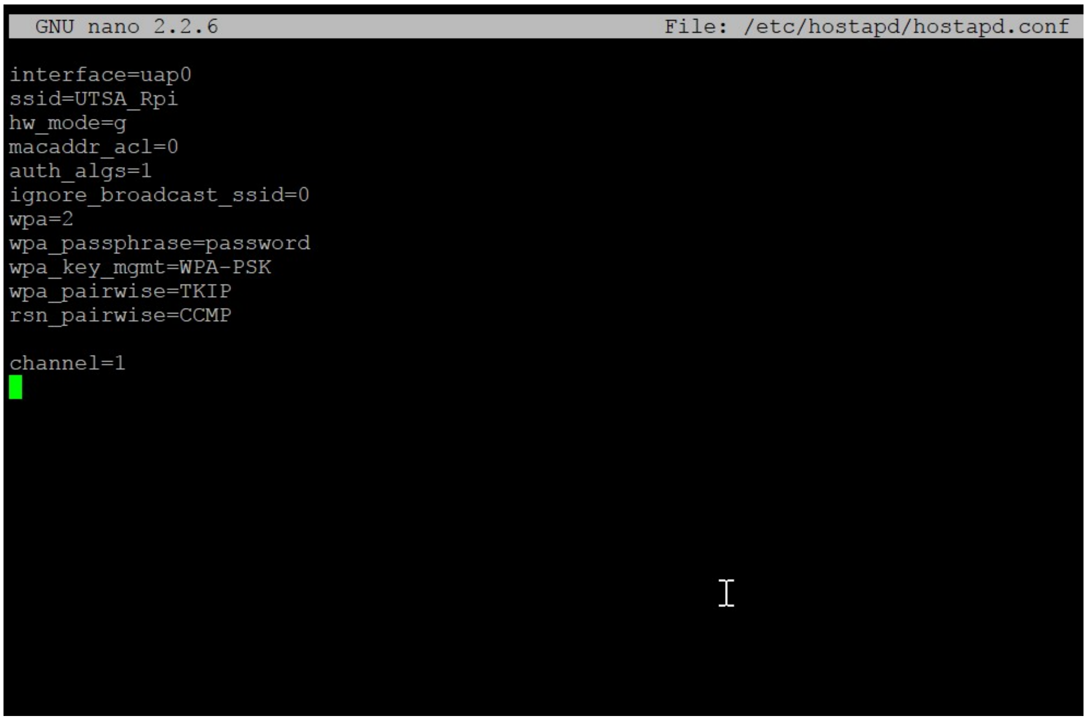
1. Raspberry Pi 3 and power adapter
2. Micro SD card for porting Linux (8GB)
3. HDMI cable and compatible monitor (for development purpose)
4. USB Keyboard and USB mouse (for development purpose)
5. Wi-Fi Router (for Demo purpose)
6. LED lights (for indication purpose)

Software and services installed on Raspberry Pi:

1. Hostapd: Linux service for hotspot interface configuration
2. Java: for backend code development on tomcat server.
3. Python: To check internet connectivity and setting of LEDs.
4. Tomcat server: Debian version of tomcat 8 is installed and deployed for hosting webpage.

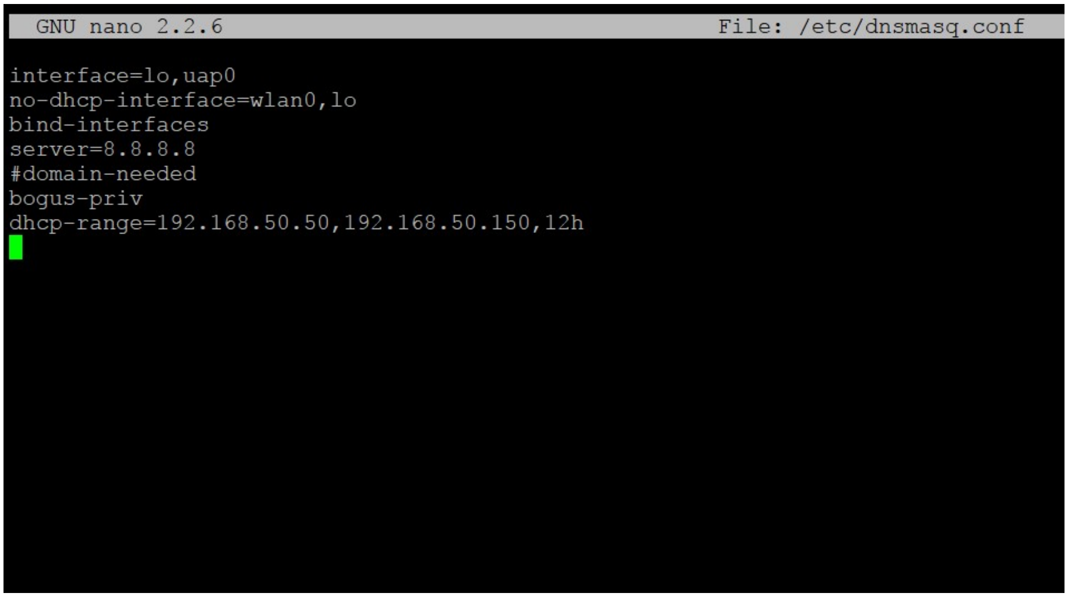
3. Services:

I. **Hostapd:** This service is installed on Raspberry Pi to create hotspot. Rpi broadcast SSID as “Utsa_Rpi” after creating uap0 interface at boot up stage. For development purpose, password is kept as “password”. Hostapd is a user space daemon for access point and authentication servers. It implements IEEE 802.11 access point management, IEEE 802.1X/WPA/WPA2/EAP Authenticators, RADIUS client, EAP server, and RADIUS authentication server. The current version supports Linux (Host AP, madwifi, mac80211-based drivers) and FreeBSD (net80211). hostapd runs in the background and acts as the backend component controlling authentication. hostapd supports separate frontend programs and an example text-based frontend, hostapd_cli, is included with hostapd.



```
  GNU nano 2.2.6                                    File: /etc/hostapd/hostapd.conf

interface=uap0
ssid=UTSA_Rpi
hw_mode=g
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=password
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

channel=1
```

II. **Dnsmasq:** This provides network infrastructure for small networks: DNS, DHCP, router advertisement and network boot. It is designed to be lightweight and have a small footprint, suitable for resource constrained routers and firewalls. It has also been widely used for tethering on smartphones and portable hotspots, and to support virtual networking in virtualization frameworks.



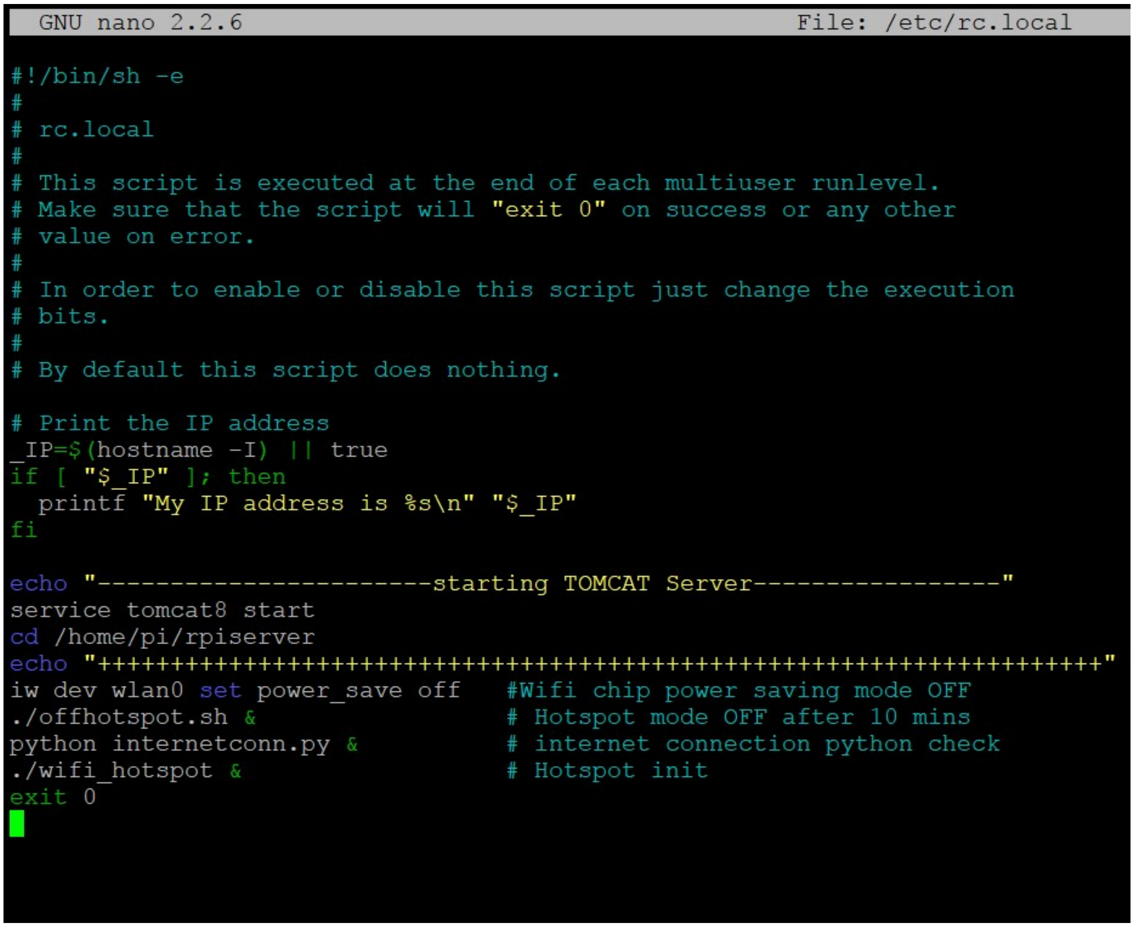
```
  GNU nano 2.2.6                                  File: /etc/dnsmasq.conf

interface=lo,uap0
no-dhcp-interface=wlan0,lo
bind-interfaces
server=8.8.8.8
#domain-needed
bogus-priv
dhcp-range=192.168.50.50,192.168.50.150,12h
```

III. **Apache Tomcat 8 webserver:** Apache Tomcat is a Servlet/JSP container and version 8.0 implements the Servlet 3.1 and JavaServer Pages 2.3 specifications. Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations.

tomcat directories:

- **/bin** - Startup, shutdown, and other scripts. The *.sh files (for Unix systems) are functional duplicates of the *.bat files (for Windows systems). Since the Win32 command-line lacks certain functionality, there are some additional files in here.
- **/conf** - Configuration files and related DTDs. The most important file in here is server.xml. It is the main configuration file for the container.
- **/logs** - Log files are here by default.
- **/webapps** – location for webapps.

IV. **/etc/rc.local:** The script `/etc/rc.local` is for use by the system administrator. It is traditionally executed after all the normal system services are started, at the end of the process of switching to a multiuser runlevel.



```
  GNU nano 2.2.6                                          File: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

echo "-----------------------starting TOMCAT Server-----------------"
service tomcat8 start
cd /home/pi/rpiserver
echo "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
iw dev wlan0 set power_save off   #Wifi chip power saving mode OFF
./offhotspot.sh &                 # Hotspot mode OFF after 10 mins
python internetconn.py &          # internet connection python check
./wifi_hotspot &                  # Hotspot init
exit 0
```

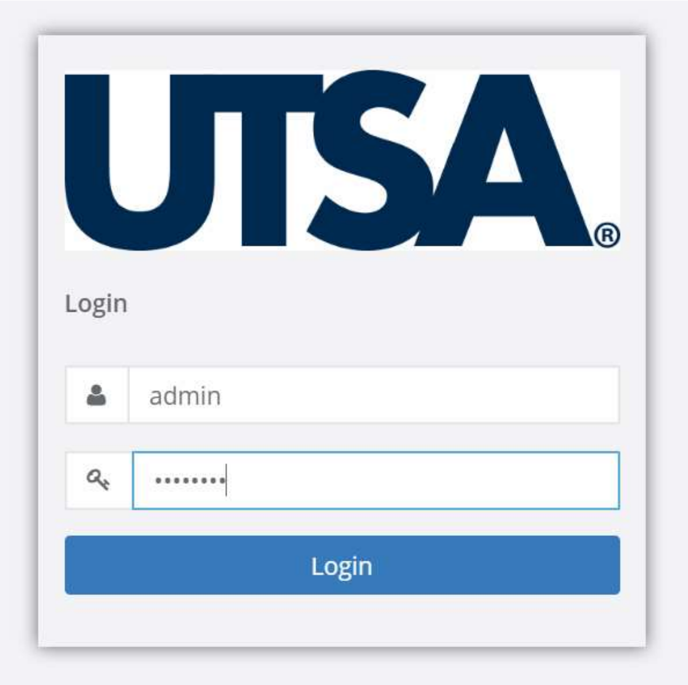
4. Internet Connectivity check (Python) and LED:

A simple python code is written to check internet connectivity by pinging “www.google.com”. This code file name is inserted in /etc/rc.local for execution after boot up.

Red Led: Shows Rpi is Offline (Not connected to internet)

Green Led: Shows Rpi is Online (Connected to internet)

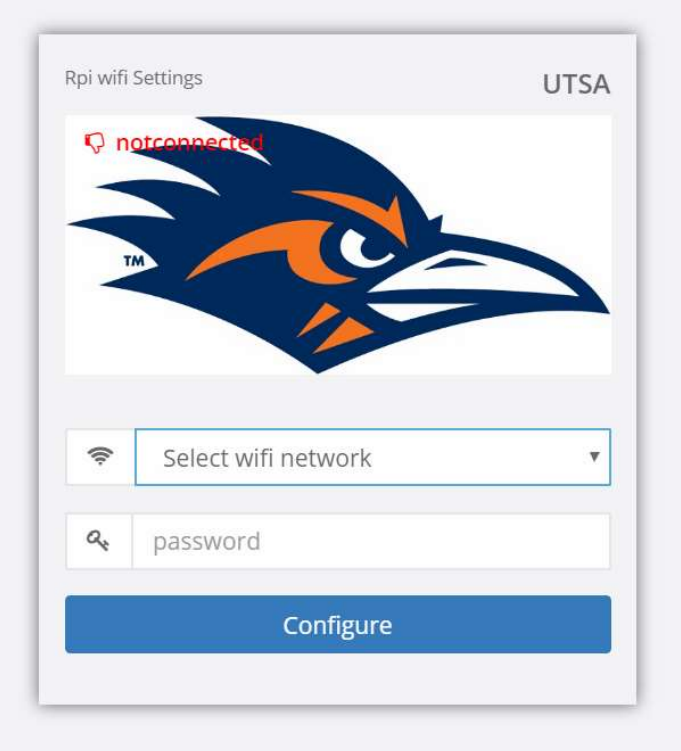
5. Screenshots:



Login

admin

••••••••

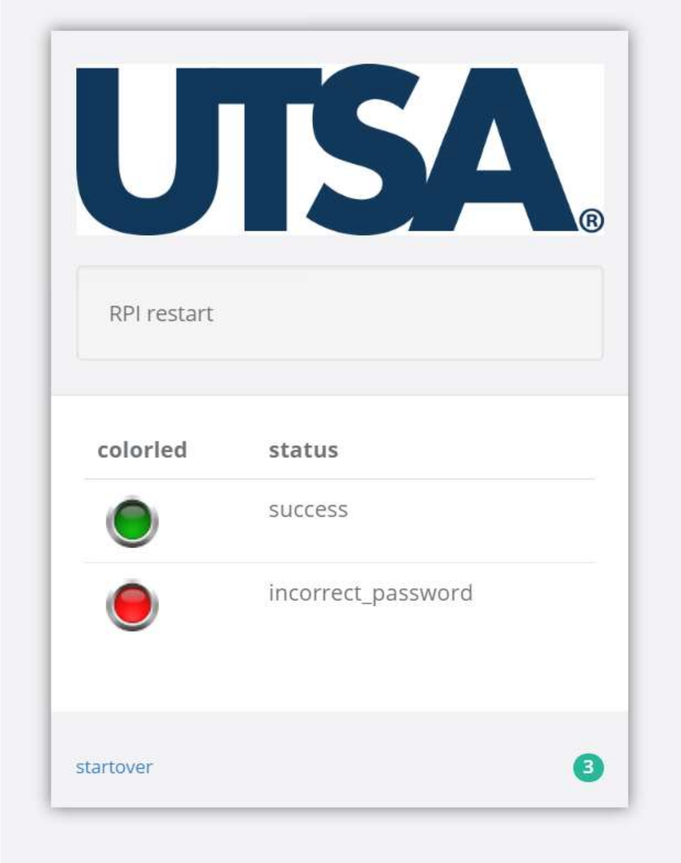


Rpi wifi Settings

notconnected

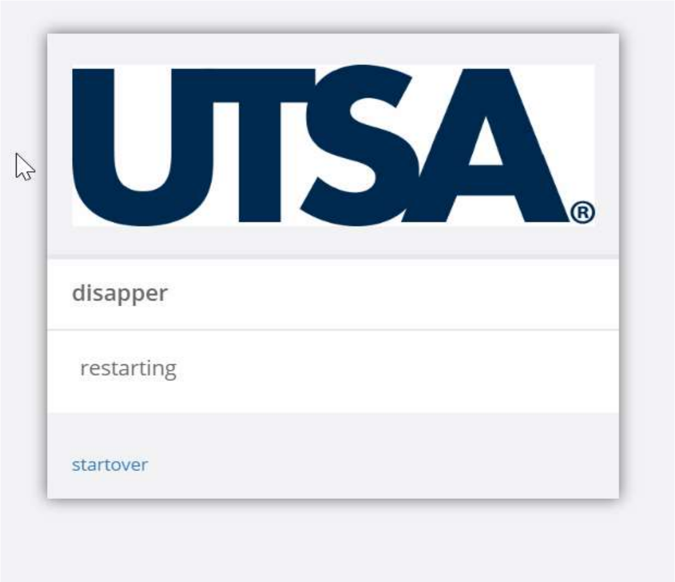
Select wifi network ▾

password



RPI restart

colorled	status
🟢	success
🔴	incorrect_password



disapper

restarting

6. Conclusion:

The latest Raspberry Pi 3 contains Bluetooth, Wi-Fi interface, LAN port, USB ports with very less price. This board is a great platform for building Internet of Things. This project facilitates setting of Wi-Fi SSID and password using mobile/tablet where getting HDMI monitor and keyboard is not that feasible on site. This project is being developed using Java and Html combination. We also used Python language to show user status of internet connectivity. With this starting point, we can add more functionality to code to make Rpi more sophisticated in IOT environment.