

# Hotspot con Raspberry Pi

Nei vari progetti che utilizzano la scheda Raspberry, può capitare di avere la necessità di far fare da Hotspot WiFi alla stessa scheda, come ad esempio quando non si ha rete WiFi disponibile.

Di seguito vedremo come attivare questo servizio e realizzeremo una semplice applicazione come ad esempio la possibilità di inviare delle scritte scorrevoli sulla scheda SenseHat presente sulla Raspberry.

## PARTE 1 – ATTIVAZIONE HOTSPOT

Prima di installare i pacchetti necessari, occorre sempre eseguire l'aggiornamento dell'elenco dei pacchetti disponibili ed il successivo aggiornamento degli stessi pacchetti, da terminale andremo a digitare i seguenti comandi:

- **sudo apt update** *aggiorna l'elenco dei pacchetti disponibili, non installa nulla.*
- **sudo apt upgrade -y** *aggiorna i pacchetti installati alla versione più recente, l'opzione -y serve per rispondere automaticamente yes a tutte le richieste ricevute durante l'installazione.*

### Approfondimento - cosa sono i REPOSITORY?

Un repository è un server (o insieme di server) che ospita pacchetti software e le informazioni necessarie per scaricarli, verificarli e installarli. Quindi non contiene solo i pacchetti .deb, ma anche:

- l'elenco dei pacchetti disponibili
- le versioni
- le dipendenze (librerie e altri pacchetti necessari al funzionamento del software)
- le firme di sicurezza (GPG) sono firme digitali crittografate che garantiscono l'autenticità (la provenienza) e l'integrità del pacchetto installato.

Per i sistemi fuori supporto EOL, cioè le versioni di S.O. che non ricevono più aggiornamenti, ci possono essere degli errori sul comando **apt update**, in questo caso è possibile risolvere il problema seguendo i seguenti passi:

1. editare il file **sources.list** con il comando **sudo nano /etc/apt/sources.list** e sostituire tutto il contenuto con il seguente: **deb [trusted=yes] http://archive.debian.org/debian buster main contrib non-free**  
Questo file dice al comando **apt**, da quali repository (server) scaricare gli aggiornamenti del sistema Debian.
2. editare il file **raspi.list**, contenente il repository, con il comando:

**sudo nano /etc/apt/sources.list.d/raspi.list** sostituire tutto il contenuto con il seguente:  
**deb http://archive.raspberrypi.org/debian buster main**

La Raspberry Pi Foundation fornisce software che Debian NON include, come:

- kernel ottimizzato per Raspberry
- firmware GPU, driver Wi-Fi / Bluetooth
- tool specifici (raspi-config, vcgencmd, ecc.)

Tutto questo non sta nei repository Debian, ma sta nel repository indicato in **raspi.list**.

3. Accettare il cambio di repository con il comando:

**sudo apt update --allow-releaseinfo-change --allow-insecure-repositories**

la prima parte **--allow-releaseinfo-change** serve per dire al comando **apt** di accettare eventuali modifiche di informazioni (versioni ecc...). La seconda parte **--allow-insecure-repositories** consente l'utilizzo di repository non firmati o con firme non valide, come nel caso di vecchie distribuzioni o repository archiviati.

4. Controllare che non ci siano altri file .list oltre al file **raspi.list** con il comando: **ls /etc/apt/sources.list.d/**  
Se ci dovessero essere altri file commentarli o rimuoverli solo se non necessari.

5. Disabilita il controllo "date scadute" sui repository archiviati editando il seguente file con il comando:

**sudo nano /etc/apt/apt.conf.d/99no-check-valid-until**

inserire all'interno del file quanto segue: **Acquire::Check-Valid-Until "false";**

6. Pulire la cache APT con i seguenti comandi: **sudo apt clean** e **sudo apt autoclean**

### Installazione pacchetti necessari.

- **sudo apt install hostapd dnsmasq -y** installa 2 pacchetti, **hostapd** per creare l'AP wifi e **dnsmasq** per assegnare l'IP dinamicamente ai dispositivi connessi
  - **sudo systemctl stop hostapd** disabilita temporaneamente i pacchetti installati perché in
  - **sudo systemctl stop dnsmasq** seguito andremo a modificare i files di configurazione
- 

### Assegnazione indirizzo e disattivazione connessione ad altre reti.

- **sudo nano /etc/dhcpd.conf** editare il file **dhcpd.conf** inserendo l'indirizzo IP della scheda sull'interfaccia WiFi disattivando la possibilità di connettersi ad un'altra rete.

Alla fine del file **dhcpd.conf** va inserito quanto segue:

- **interface wlan0** Indica che le impostazioni successive si applicano all'interfaccia Wi-Fi
  - **static ip\_address=192.168.4.1/24** imposta l'indirizzo IP e la subnet mask a 24 bit
  - **nohook wpa\_supplicant** Impedisce a **dhcpd** di avviare automaticamente il file **wpa\_supplicant**, che normalmente gestisce la connessione a reti Wi-Fi. Serve quando vuoi usare **wlan0** come **hotspot** o **rete ad-hoc**, senza che il dispositivo provi a collegarsi a un Wi-Fi esterno.
- 

### Attivazione servizio DHCP

- **sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig** fare il backup del file **dnsmasq.conf**
- **sudo nano /etc/dnsmasq.conf** creare un nuovo file

All'interno di questo file inserire quanto segue:

- **interface=wlan0** indica che il servizio di DHCP offerto dal pacchetto **dnsmasq**, è attivo solo sull'interfaccia **wlan0**
  - **dhcp-range=192.168.4.10,192.168.4.50,255.255.255.0,24h** indica il range di indirizzi IP assegnabili, la loro **subnetmask** e la durata del **lease**, cioè il tempo per quanto tempo un indirizzo IP rimane assegnato ad un **Host**, in questo caso 24 ore.
-

## Configurazione Hotspot

- **sudo nano /etc/hostapd/hostapd.conf**

*editare il file **hostapd.conf***

All'interno di questo file inserire quanto segue:

- **interface=wlan0** *specifica l'interfaccia di rete da usare come Access Point*
- **driver=nl80211** *indica il driver da utilizzare per la scheda WiFi*
- **ssid=Raspberry\_AP** *indica il nome della rete (SSID)*
- **hw\_mode=g** *specifica lo standard WiFi da utilizzare, g=802.11g 2,4Ghz*
- **channel=7** *indica il canale da utilizzare*
- **wmm\_enabled=0** *WMM (Wi-Fi Multimedia) gestisce priorità del traffico (voce, video, dati) 0 disabilita WMM*
- **macaddr\_acl=0** *controllo accessi tramite MacAddress 0= nessun controllo*
- **auth\_algs=1** *algoritmo di autenticazione accettato, se 1 usa **Open System authentication***
- **ignore\_broadcast\_ssid=0** *se 0 la rete è visibile, se 1 la rete è nascosta*
- **wpa=2** *tipo di sicurezza WPA da utilizzare*
- **wpa\_passphrase=raspberrry123** *indica la password, minimo 8 caratteri*
- **wpa\_key\_mgmt=WPA-PSK** *tipo di gestione delle chiavi WPA , PSK = Pre Shared Key cioè password condivisa*
- **rsn\_pairwise=CCMP** *tipo di algoritmo di cifratura CCMP basato su AES*

- **sudo nano /etc/default/hostapd**

*editare il file **hostapd** per collegare il pacchetto che gestisce l'Access Point al file editato prima **hostapd.conf**.*

All'interno di questo file inserire quanto segue:

**DAEMON\_CONF="/etc/hostapd/hostapd.conf"**

**hostapd** è un programma e occorre indicargli il file di configurazione da usare.

---

## Abilitazione dei servizi e riavvio

- **sudo systemctl unmask hostapd** *disabilita il blocco dell'AP su alcune versioni di S.O.*
  - **sudo systemctl enable hostapd** *abilita l'avvio automatico del pacchetto.*
  - **sudo systemctl enable dnsmasq** *abilita l'avvio automatico del pacchetto.*
  - **sudo reboot** *riavvio.*
-

## PARTE 2 – REALIZZAZIONE DI UN WEBSERVER SU RASPBERRY CON AP

La struttura del sistema è la stessa vista nell'esercizio al seguente tutorial:

<https://danielepostacchini.it/wp-content/uploads/2020/02/Esercizio-webserver-Raspberry.pdf>

Come nell'esempio utilizzeremo anche questa volta una scheda Raspberry con a bordo una scheda Sensehat. A differenza di quell'esempio per comodità metteremo il file html all'interno della scheda Raspberry, in modo da accedere con un Qrcode alla pagina.

Servono comunque i seguenti files:

- Pagina html con form, sul server. **scritta.html**
- Programma PHP, sul server. **controllo.php**
- File Python, sul server. **sensehat\_display.py**

Il file HTML ed il file PHP andranno collocati all'interno del percorso **/var/www/html**, in modo da accedere da remoto digitando l'indirizzo <http://192.168.4.1/scritta.html>.

Il file in **sensehat\_display.py**, lo metteremo nel Desktop, ma potrebbe essere posizionato anche altrove.

### File html con form – scritta.html

Una versione semplificata potrebbe essere la seguente, che consentirà l'inserimento di uno dei 4 colori, della velocità in msec e della scritta da visualizzare. Il form invierà 3 valori al server.

In questa versione stiamo utilizzando il modo GET, ma si potrebbe tranquillamente utilizzare anche il modo POST.

Sul client vedremo la seguente schermata.

Volendo utilizzare oltre all'HTML anche il CSS si potrebbe ottenere una versione sicuramente più accattivante e colorata. Allego insieme al tutorial entrambe le versioni.

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <title>Controllo Sense HAT</title>
</head>
<body>
  <h2>Invia testo alla Sense HAT</h2>
  <form action="http://192.168.4.1/controllo.php" method="get">
    <label for="colore">Colore (rosso, verde, blu, bianco):</label>
    <input type="text" id="colore" name="colore" required><br><br>

    <label for="velocita">Velocità (ms):</label>
    <input type="number" id="velocita" name="velocita" required><br><br>

    <label for="testo">Testo da mostrare:</label>
    <input type="text" id="testo" name="testo" required><br><br>

    <input type="submit" value="Invia">
  </form>
</body>
</html>
```

### Invia testo alla Sense HAT

Colore (rosso, verde, blu, bianco):

Velocità (ms):

Testo da mostrare:

### Invia testo alla Sense HAT

Colore

rosso, verde, blu, bianco

Inserisci uno dei colori supportati

Velocità (ms)

es. 200

Testo da mostrare

Scrivi il messaggio

**Invia alla Sense HAT**

## File PHP - controllo.php

Come nell'esempio citato sopra il file PHP farà da tramite tra il form ed il file in Python. In questo caso invierà 3 stringhe e leggerà 4 risposte da rimandare al client.

```
<?php
$colore = $_GET['colore'];
$velocita = $_GET['velocita'];
$testo = $_GET['testo'];

// Comando per eseguire lo script Python
$comando = "sudo python /home/pi/Desktop/sensehat_display.py $colore $velocita $testo";
exec($comando, $out, $status); // Lancia il comando in background

echo $out[0];
echo $out[1].<br>; //br per andare a capo
echo $out[2];
echo $out[3];
?>
```

Il file PHP legge le 3 variabili inviate in modo GET e compone una stringa **\$comando**, che conterrà il comando che manderà in esecuzione il file Python passando le 3 variabili.

La risposta del programma eseguito andrà nel vettore **out**, e 4 elementi di questo vettore verranno inviati al client con il comando **echo**.

---

## File Python - sensehat\_display.py

Nel file vengono importate le librerie necessarie, lette le variabili passate dal file php ed eseguita la funzione richiesta, leggendo i commenti si può capire facilmente il funzionamento del codice scritto.

```
import sys #Importa il modulo sys, che permette di accedere agli argomenti
            #passati al programma da riga di comando (sys.argv)
from sense_hat import SenseHat #Importa la classe SenseHat per controllare la scheda
import time # Importa il modulo time, non usato nel programma

sense = SenseHat() #Crea l'oggetto SenseHat per comunicare con la Sense HAT

colore_input = sys.argv[1] #Legge il primo argomento passato da riga di comando
velocita = int(sys.argv[2]) #Legge il secondo argomento e lo converte in intero
testo = sys.argv[3] #Legge il terzo argomento passato al programma

if colore_input == "rosso": #Se l'utente scrive "rosso", imposta il colore rosso
    colore_rgb = (255, 0, 0)
elif colore_input == "verde": #Se l'utente scrive "verde", imposta il colore verde
    colore_rgb = (0, 255, 0)
elif colore_input == "blu": #Se l'utente scrive "blu", imposta il colore blu
    colore_rgb = (0, 0, 255)
elif colore_input == "bianco": #Se l'utente scrive "bianco", imposta il colore bianco
    colore_rgb = (255, 255, 255)
else:
    colore_rgb = (255, 255, 255) #Se il colore inserito non è riconosciuto, usa il bianco

# Mostra testo scorrevole e converte la velocità da millisecondi a secondi
sense.show_message(testo, scroll_speed=velocita / 1000.0, text_colour=colore_rgb)
print("scritta eseguita=" # Stampa un messaggio di conferma a console
print(testo) # Stampa il testo che è stato mostrato
print("colore=") # Stampa l'etichetta del colore
print(colore_input) # Stampa il colore scelto dall'utente
```

## TEST

A questo punto per provare il sistema realizzato è sufficiente copiare i due files scritta.html e controllo.php nel percorso /var/www/html della scheda Raspberry ed il file sensehat\_display.py sul Desktop.

Per rendere tutto più agevole si può creare un Qrcode dell'url <http://192.168.4.1/scritta.html> (ci sono diverse app gratuite per creare Qrcode da un testo).

Per utilizzare il sistema realizzato sarà sufficiente connettersi alla WiFi creata dalla scheda Raspberry ed inquadrare il Qrcode, in questo modo si aprirà la pagina html con il form.

**Buon divertimento!**



danielepostacchini.it