

IOTAIUTO

PANORAMICA E PRINCIPI GENERALI

```
# LA STRUTTURA DI QUESTA GUIDA SI ARTICOLA IN:
```

```
# 1. risorse HW utilizzate
```

```
# 2. passi necessari a rendere operativa ciascuna di esse
```

```
# IL ROUTER ANDRA' CONFIGURATO AFFINCHE' A OGNI DEVICE SIA ASSEGNATO UN IP-ADDRESS-FISSO
```

```
# NEL RANGE 192.168.1.221 - 192.168.1.240
```

RASPBERRY pi 3 B+

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

SCHEDA MICRO SD (ALMENO CLASSE-10)

https://www.amazon.it/SanDisk-MicroSDHC-Adattatore-Performance-Velocit%C3%A0/dp/B073K14CVB/ref=sr_1_3?qid=1560245612&refinements=p_n_feature_browse-bin%3A516697031&s=electronics&sr=1-3

PREPARAZIONE SCHEDA MICRO SD: INSTALLAZIONE O.S. "RASPBIAN LITE"

```
# 1. prima di procedere, entrare come ADMIN nel ROUTER e assegnare l IP STATICO 192.168.1.221 al Mac-address del RASPBERRY
```

```
# 2. download OS
```

<https://www.raspberrypi.org/downloads/raspbian/>

```
# formattazione micro sd
```

```
# scaricare balenaEtcher poi eseguirlo per montare OS
```

```
# su micro sd
```

<https://www.balena.io/etcher/>

```
# inserire micro sd nel RASPBERRY e alimentarlo

# configurare il RASPBERRY:
sudo raspi-config

# nella configurazione che comparira' a video, abilitare:
# 1. connessione ssh

# 2. wifi o Ethernet

# poi aprire una sessione terminale per

# configurare il NOME-MACCHINA e la relativa PASSWORD O.S. sostituendo:
# "raspberrypi" con "ha-xxxxxxxx" - ossia con la VPN-CLIENTE - affinche
# NOME-MACCHINA e VPN-CLIENTE corrispondano

sudo nano /etc/hostname
sudo nano /etc/hosts
sudo hostname ha-xxxxxxxx
sudo reboot
```

INSTALLAZIONE PYTHON3 + HOME-ASSISTANT + RELATIVO SERVIZIO ABILITATO sul O.S.

```
# 1. installare python3 e relativi componenti a corredo

sudo apt install python3 python3-venv incron mc python3-dev libffi-dev
libssl-dev
python3 -m venv ~/lib
~/lib/bin/activate
echo "source ~/lib/bin/activate" >> ~/.bashrc

# 2. installare l'applicazione HOMEASSISTANT

pip install -U pip
pip install homeassistant

# 3. creare con NANO il servizio HomeAssistant nel file:

# /etc/systemd/system/home-assistant.service

[Unit]
Description=Home Assistant
After=network.target
```

```
[Service]
Type=simple
User=pi
ExecStart=/home/pi/lib/bin/hass -c "/home/pi/.homeassistant"

[Install]
WantedBy=multi-user.target

# 4. abilitare il servizio Homeassistant

systemctl daemon-reload
systemctl enable home-assistant
```

CREAZIONE + INSTALLAZIONE + ATTIVAZIONE servizio "OPENVPN" linux client + relativo CERTIFICATO

```
# 1. ricordarsi di aggiornare il file users.yaml sul server di iotaiuto.it con il nuovo utente e relativo ipaddr
# 2. eseguire poi i comandi "sctl restart cloud" + "sctl status cloud" per recepire il nuovo utente
# 3. chiedere a Igor di inserire il nuovo utente ed il suo nuovo ipaddr nel file dove sono censiti i DNS operanti
#     in iotaiuto.it
```

```
<code>
apt-get install openvpn
```

/etc/openvpn/client1.conf

```
dev tun
proto tcp-client

remote legnago.csgalileo.org 1194

ca  /etc/easy-rsa-legnago/keys/ca.crt
cert /etc/easy-rsa-legnago/keys/nms.crt
key  /etc/easy-rsa-legnago/keys/nms.pem

tls-client
port 1194

user nobody
group nogroup

#comp-lzo # Do not use compression. It doesn't work with RouterOS (at least up to RouterOS 3.0rc9)

# More reliable detection when a system loses its connection.
```

```

ping 15
ping-restart 45
ping-timer-rem
persist-tun
persist-key

# Silence the output of replay warnings, which are a common false
# alarm on WiFi networks. This option preserves the security of
# the replay protection code without the verbosity associated with
# warnings about duplicate packets.
mute-replay-warnings

# Verbosity level.
# 0 = quiet, 1 = mostly quiet, 3 = medium output, 9 = verbose
verb 3

cipher AES-256-CBC
auth SHA1
pull

auth-user-pass auth.cfg
script-security 2
up /etc/openvpn/up.sh

```

/etc/openvpn/up.sh (chmod +x)

```

#!/bin/sh

ip route add 10.90.0.0/16 via 10.15.32.33

```

/etc/openvpn/auth.cfg

```

username
password

```

Start service with systemd

```

systemctl start openvpn@client1
systemctl enable openvpn@client1

```

il certificato da usare per il cliente si trova al seguente link: [Clienti HA](#)

BROADLINK RM Mini3 / RM 2

[https://www.amazon.it/Snowsound-SmartHome-infrarossi-Broadlink-Telefono/dp/B076TGXRWS/ref=sr_1_3?__mk_it_IT=%C3%85M%C3%85%C5%BD%C3%95%C3%91&cid=2Q9SIRMA3OUT4&keywords=broadlink+rm+mini3&qid=1560415414&s=gateway&suffix=broadlin%2Caps%2C219&sr=8-3](https://www.amazon.it/Snowsound-SmartHome-infrarossi-Broadlink-Telefono/dp/B076TGXRWS/ref=sr_1_3?__mk_it_IT=%C3%85M%C3%85%C5%BD%C3%95%C3%91&cid=2Q9SIRMA3OUT4&keywords=broadlink+rm+mini3&qid=1560415414&s=gateway&sprefix=broadlin%2Caps%2C219&sr=8-3)

CONFIGURAZIONE BROADLINK RM Mini3 /RM2

```
# 1.2 Long press the reset button until the blue LED is blinking quickly.  
# 1.3 Long press again until blue LED is blinking slowly.  
# 1.4 Manually connect to the WiFi SSID named BroadlinkProv.  
# 1.5 Run setup() and provide your ssid, network password (if secured),  
and set the security mode  
# 1.6 Security mode options are (0 = none, 1 = WEP, 2 = WPA1, 3 = WPA2, 4  
= WPA1/2)  
  
# 2. la configurazione del dispositivo per accedere alla rete WIFI domestica  
si esegue entrando  
# nella modalità TERMINALE di RASPBIAN e svolgendo i seguenti passi:  
# 2.1 installazione applicazione BROADLINK con versione di crittografia  
0.10 (obbligatoria altrimenti  
# non funziona)  
# 2.2 configurazione dispositivo BROADLINK perchè acceda alla rete WIFI  
domestica  
  
pip install broadlink==0.10  
pip install pycrypto  
  
import broadlink  
  
broadlink.setup('myssid', 'mynetworkpass', 3)  
  
# 3. poi si preleveranno dal repository GITHUB:  
# 3.1 il programma che permette al BROADLINK di operare come CLIENT  
# 3.2 il programma che permette al BROADLINK di eseguire la funzione  
DISCOVERY  
# 3.3 quindi si procede a renderli eseguibili  
wget https://raw.githubusercontent.com/mjg59/python-  
broadlink/master/cli/broadlink_cli
```

```
 wget https://raw.githubusercontent.com/mjg59/python-broadlink/master/cli/broadlink_discovery
```

```
 chmod +x broadlink_cli broadlink_discovery
```

LEARNING CODICI I.R. CON BROADLINK RM Mini3/rm2

```
# Sempre operando da TERMINALE del RASPBERRY1, ora si farà in modo che il BROADLINK possa "imparare" i codici
```

```
# I.R. o R.F. emessi dai rispettivi telecomandi dei vari apparecchi (TV, CLIMATIZZATORE, MEDIAPLAYER, ecc.);
```

```
# si esegue quindi il seguente comando, che identifica i devices BROADLINK connessi:
```

```
./broadlink_discovery
```

```
# esempio dell'output del "discovery":
```

```
broadlink_cli --type 0x2787 --host 192.168.2.28 --mac a126f434ea34
```

```
# si esegue quindi il seguente comando, che pone il BROADLINK in stato di "apprendimento" IR oppure RF:
```

```
__**apprendimento IR**__
```

```
./broadlink_cli --type 0x2787 --host 192.168.2.28 --mac a126f434ea34 --learn
```

```
# a terminale comparirà una stringa di caratteri che rappresenta il comando emesso dal telecomando;
```

```
# quella stringa è già in formato "base64" e come tale pronta ad essere inserita nel corrispondente file
```

```
# switch.yaml sotto l'etichetta del tasto/funzione del telecomando cui è riferita
```

```
__** apprendimento RF **__
```

```
./broadlink_cli --type 0x2787 --host 192.168.2.28 --mac a126f434ea34 --rfscan

# a terminale comparirà una stringa di caratteri che rappresenta il comando emesso dal telecomando;

# quella stringa N-0-N è in formato "base64" e perciò deve essere prima convertita in tale formato;

# per la sua conversione, a terminale si eseguirà il seguente comando di codifica:

echo 'xyzaaabbbccc123456' | base64

# la stringa restituita andrà poi inserita nel corrispondente file switch.yaml

# sotto l'etichetta del tasto/funzione del telecomando cui è riferita

# per eseguire quanto sopra illustrato, si azionerà il tasto di ACCENSIONE del telecomando (puntato

# sul BROADLINK) e così di seguito per VOLUME+ VOLUME- CANALE+ CANALE- e da CANALE-1 a CANALE-NNN

# il file che occorre aggiornare si trova in: /homeassistant/switch.yaml
```

Creazione Virtualhost

Esempio pratico: voglio creare un punto di ingresso per <http://robby.iotaiuto.it>

In /etc/nginx/vpn-enabled relativamente a questo esempio, creare un file: roby.conf

```
# Questo è l'insieme di regole che gestiscono http://robby.iotaiuto.it
server {
    # PUNTO 1
    server_name roby.iotaiuto.it;
    listen 80;
    return 301 https://$host$request_uri;
}

# Questo è l'insieme di regole che gestiscono https://robby.iotaiuto.it
```

```

server {
    listen 443 ssl http2;
    # PUNTO 2
    server_name roby.iotaiuto.it;

    ssl_certificate
    /etc/letsencrypt/live/iotaiuto.it.wildcard/fullchain.pem;
    ssl_certificate_key
    /etc/letsencrypt/live/iotaiuto.it.wildcard/privkey.pem;
    ssl_dhparam /etc/nginx/ssl/dhparams.pem;

    add_header Strict-Transport-Security "max-age=31536000;
includeSubdomains";
    ssl on;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers
"EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:!aNULL:!eNULL:!EXPORT:!DES:
!MD5:!PSK:!RC4";
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;

    proxy_buffering off;

    location / {
        # PUNTO 3
        proxy_pass http://10.44.98.13:8123;
        proxy_set_header Host $host;
        proxy_redirect http:// https://;
        proxy_http_version 1.1;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
    }
}

```

Dopo di che eseguire un check della configurazione

```

root@homeassistant:/etc/nginx/vpn-enabled# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

```

e in caso di **ok** riavviare nginx, perché acquisisca il nuovo endpoint.

```

root@homeassistant:/etc/nginx/vpn-enabled# systemctl restart nginx
root@homeassistant:/etc/nginx/vpn-enabled# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor
   preset: enabled)
     Active: active (running) since mar 2020-08-18 13:50:00 CEST; 59min ago
           ^^^^^^^^^^^^^^^^^^

```

Nel PUNTO 1 e PUNTO 2 va la parte dell'url che segue i due slash nell'indirizzo, in questo esempio da <http://roby.iotaiuto.it> resta solo roby.iotaiuto.it

Nel PUNTO 3 va l'IP con cui il raspberry si collega alla VPN

From:
<https://wiki.csgalileo.org/> - **Galileo Labs**

Permanent link:
<https://wiki.csgalileo.org/projects/internetofthings/iotaiuto>

Last update: **2020/08/21 10:31**

