

# Installazione HotSpot DMR con gateway DMR ↔ D-STAR

(Antonio Matraia, [IU5JAE](#))

## Scopo

Lo scopo di queste note è quello di configurare un hotspot casalingo da impiegarsi per la connessione alla rete DMR (in questo caso brandmeister) e, tramite l'XLX039, di poter accedere anche alla rete D-Star.

La soluzione proposta, partendo da una distribuzione “pulita” ha il vantaggio, rispetto a soluzioni molto più semplici da configurare (esempio pi-star, easyBM, ecc.), che può essere facilmente adattata a far coesistere altri usi del raspberry con quello dell'hotspot (nel mio caso viene impiegato anche come mediacenter tramite Kodi), al prezzo di un'installazione e di una configurazione più complicata.

Tutta l'installazione e la configurazione verrà eseguita da riga di comando<sup>1</sup>, senza l'installazione di nessuna interfaccia grafica.

Buon divertimento !!

## Materiale impiegato

### Hardware<sup>2</sup>

- Raspberry pi 3 (attenzione al case che deve essere abbastanza alto da permettere l'alloggiamento della DVMEGA)
- Scheda SD card (in questo caso da 32 GB)
- Alimentatore 2 A
- Scheda DVMEGA (con FW HR3.14) + antenna

### Software<sup>3</sup>

- [Raspbian Stretch Lite](#)
- [MMDVMHost](#)
- [MMDVMDashboard](#)
- [DMRGateway](#)

## Operazioni preliminari

Dopo aver scaricato l'immagine, scrivere la SD ad esempio utilizzando Win32DiskImager e ricordarsi, se vogliamo usare l'accesso ssh, di creare un file (anche vuoto) nella cartella **/boot** con nome **ssh**, altrimenti al primo boot non sarà possibile collegarsi con un client ssh.

---

1 Tutti i comandi sono in carattere [monospazio](#) di colore blu

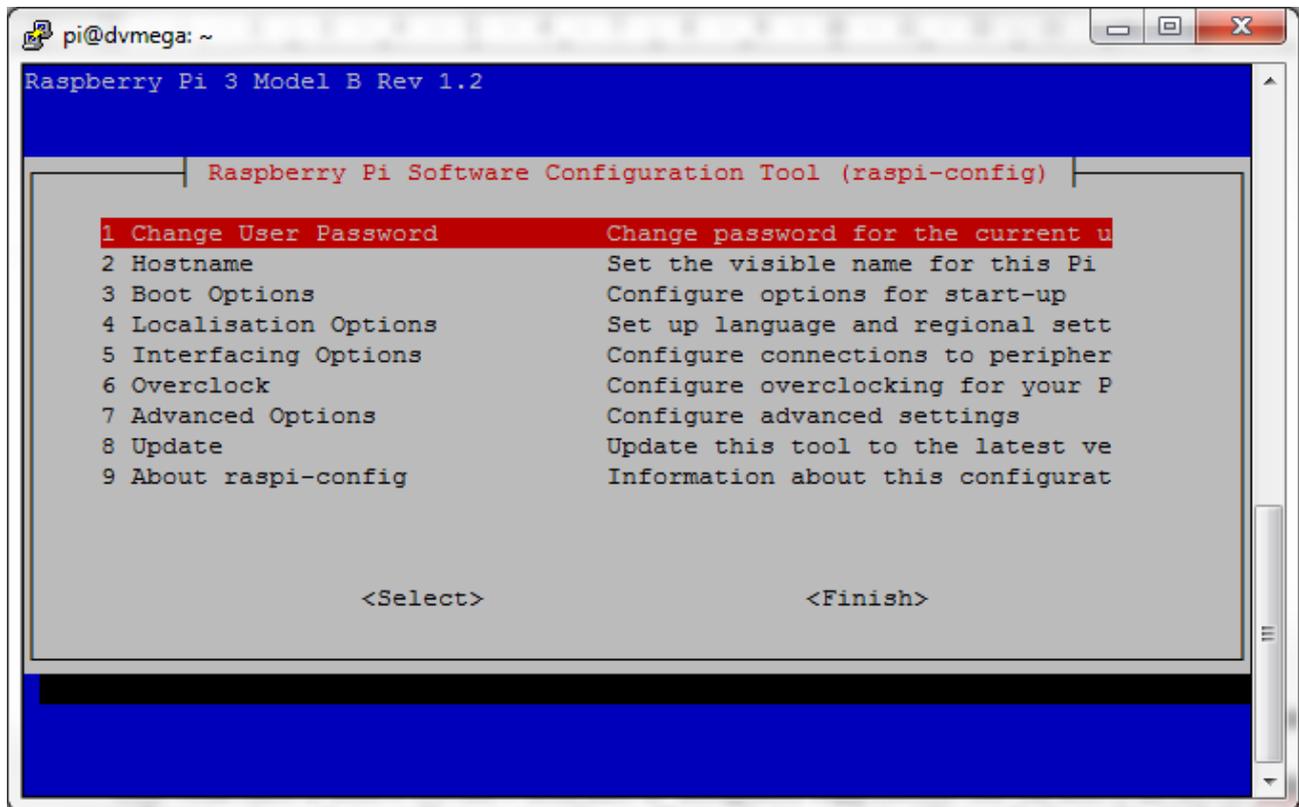
2 Tutte le prove sono state eseguite con una radio TYT MD380

3 Software scaricato dal 20/09 al 25/09 2017

Una volta fatto il boot e trovato l'indirizzo IP assegnato<sup>4</sup>, supponendo che sia attivo un server DHCP sulla rete, ci possiamo collegare usando un client ssh (es. putty) utilizzando le credenziali di default pi/raspberrypi, una volta collegati queste sono le prime configurazioni da eseguire:

```
sudo raspi-config
```

e nella schermata sotto riportata configurare almeno:



- Hostname<sup>5</sup>
- Lingua e timezone (in Localisation Options)
- Expand filesystem (in Advanced Options)
- Eventuali configurazioni richieste per altri scopi

Completate queste impostazioni eseguire il reboot, come suggerito.  
Al riavvio aggiornare tutti i pacchetti installati con i seguenti comandi:

- 4 In questo documento si considera di utilizzare solo l'interfaccia di rete "wired", se vogliamo disabilitare l'interfaccia wireless, aggiungere la seguente riga al file /boot/config.txt: dtoverlay=pi3-disable-wifi
- 5 Se vogliamo usare un indirizzo IP fisso, con le nuove versioni di raspbian è cambiato il modo di configurazione, dobbiamo editare il file /etc/dhcpd.conf (ad esempio con il comando `sudo vi /etc/dhcpd.conf`) ed inserire in fondo queste righe (il nome dell'interfaccia può essere trovato con il comando `ifconfig`) con i parametri della rete che vogliamo configurare:

```
interface enxb827eb5d234a
static ip_address=192.168.2.9/24
static routers=192.168.2.1
static domain_name_servers=8.8.8.8 208.67.220.220
```

altrimenti possiamo impostare un indirizzo riservato abbinato al MAC della scheda di rete sul server DHCP (soluzione più "elegante" soprattutto se utilizziamo l'hotspot su più reti).

```
sudo apt update
```

e

```
sudo apt upgrade
```

dopo un altro reboot tramite il comando:

```
sudo reboot
```

il sistema è pronto per installare il software (nel mio caso a questo punto ho installato tutta la parte riguardante l'uso come mediacenter), possiamo installare due software che utilizzeremo in seguito (git e screen):

```
sudo apt-get install git screen
```

Passiamo adesso all'installazione della parte riguardante l'hotspot vero e proprio secondo questi passi:

1. Installazione e configurazione di MMDVMHost per il collegamento diretta alla sola rete DMR brandmeister.
2. Installazione e configurazione di MMDVMDashboard (interfaccia web per il controllo di MMDVMHost).
3. Test del corretto funzionamento dell'accesso alla rete brandmeister.
4. Installazione e configurazione di DMRGateway per l'accesso sia alla rete brandmeister che alla rete D-Star attraverso il reflector XLX039.
5. Modifica della configurazione di MMDVMHost per l'uso di DMRGateway.
6. Test del corretto funzionamento dell'accesso alle due reti configurate.

## Installazione MMDVMHost

Questo è il software "cuore" del nostro hotspot, la sua compilazione è semplice e viene eseguita, usando i sorgenti aggiornati presenti su github, con questi comandi:

```
cd /opt
```

```
sudo git clone https://github.com/g4klx/MMDVMHost.git
```

```
cd /opt/MMDVMHost
```

```
sudo make
```

se tutto è andato a buon fine non resta che disabilitare tutti i servizi legati all'interfaccia seriale integrata nel connettore GPIO ed editare il file di configurazione.

Per disabilitare l'interfaccia seriale arrestare il servizio:

```
sudo systemctl stop serial-getty@ttyAMA0.service
```

disabilitare il riavvio al boot:

```
sudo systemctl disable serial-getty@ttyAMA0.service
```

editare il file **/boot/cmdline.txt** ed eliminare tutti i riferimenti a **/dev/ttyAMA0** o **serial0**:

```
sudo nano /boot/cmdline.txt
```

questo il file originale con barrata la parte da eliminare:

```
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=PARTUUID=ef32eb07-02 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait
```

ed, in caso di Raspberry pi 3, disabilitare anche l'interfaccia integrata Bluetooth:

```
sudo bash -c 'echo "dtoverlay=pi3-disable-bt" >> /boot/config.txt'
```

e quindi riavviamo:

```
sudo reboot
```

Al riavvio rimane solo da editare il file **MMDVM.ini** con i comandi:

```
cd /opt/MMDVMHost
```

e

```
sudo nano MMDVM.ini
```

questo il file testato (in rosso i parametri da personalizzare):

```
[General]
Callsign=xxxxxxx6
Id=xxxxxxx7
Timeout=240
Duplex=0
# ModeHang=10
RFModeHang=20
NetModeHang=20
Display=None
Daemon=0

[Info]
RXFrequency=4334500008
TXFrequency=433450000
Power=1
Latitude=xx.x9
Longitude=xx.x
Height=0
Location=xxxxx10
```

---

6 Inserire nominativo

7 Inserire DMR-ID

8 Inserire la frequenza dell'hospot, quella proposta è quella suggerita da dmr-italia al seguente link:  
<http://www.dmr-italia.it/frequenza-unica-per-dv4mini/>

9 Inserire latitudine (> 0 Nord) e longitudine (> 0 Est)

10 Inserire località

Description=xxxxxx<sup>11</sup>  
URL=https://qrz.com/db/xxxxxx<sup>12</sup>

[Log]  
# Logging levels, 0=No logging  
DisplayLevel=0  
FileLevel=2  
FilePath=.  
FileRoot=MMDVM

[CW Id]  
Enable=0  
Time=10  
# Callsign=

[DMR Id Lookup]  
File=DMRIds.dat  
Time=24

[Modem]  
Port=/dev/ttyAMA0  
TXInvert=1  
RXInvert=0  
PTTInvert=0  
TXDelay=100  
RXOffset=0  
TXOffset=0  
DMRDelay=0  
RXLevel=50  
TXLevel=50  
TXDCOffset=0  
# CWidTXLevel=50  
# D-StarTXLevel=50  
DMRTXLevel=50  
# YSFTXLevel=50  
# P25TXLevel=50  
RSSIMappingFile=RSSI.dat  
Trace=0  
Debug=0

[UMP]  
Enable=0  
# Port=\\.COM4  
Port=/dev/ttyACM1

[D-Star]  
Enable=0  
Module=C  
SelfOnly=0  
AckReply=1  
AckTime=750  
ErrorReply=1  
# ModeHang=10

[DMR]  
Enable=1  
Beacons=1  
ColorCode=1  
SelfOnly=0  
EmbeddedLCOnly=0

---

11 Inserire una descrizione per l'hotspot  
12 Inserire il link alla pagina QRZ (o altro a scelta)

DumpTADData=1  
# Prefixes=234,235  
# Slot1TGWhiteList=  
# Slot2TGWhiteList=  
CallHang=3  
TXHang=3  
# ModeHang=10

[System Fusion]  
Enable=0  
LowDeviation=0  
SelfOnly=0  
#DSQ=1  
RemoteGateway=0  
# ModeHang=10

[P25]  
Enable=0  
NAC=293  
SelfOnly=0  
OverrideUIDCheck=0  
# ModeHang=10

[D-Star Network]  
Enable=0  
GatewayAddress=127.0.0.1  
GatewayPort=20010  
LocalPort=20011  
# ModeHang=3  
Debug=0

[DMR Network]  
Enable=1  
Address=brandmeister.digitalham.it  
Port=62031  
Jitter=300  
# Local=62032  
Password=passw0rd  
# Options=  
Slot1=0  
Slot2=1  
# ModeHang=3  
Debug=0

[System Fusion Network]  
Enable=0  
LocalAddress=127.0.0.1  
LocalPort=3200  
GwyAddress=127.0.0.1  
GwyPort=4200  
# ModeHang=3  
Debug=0

[P25 Network]  
Enable=0  
GatewayAddress=127.0.0.1  
GatewayPort=42020  
LocalPort=32010  
# ModeHang=3  
Debug=0

[TFT Serial]

```
# Port=modem
Port=/dev/ttyAMA0
Brightness=50

[HD44780]
Rows=2
Columns=16

# For basic HD44780 displays (4-bit connection)
# rs, strb, d0, d1, d2, d3
Pins=11,10,0,1,2,3

# Device address for I2C
I2CAddress=0x20

# PWM backlight
PWM=0
PWMPin=21
PWMBright=100
PWMDim=16

DisplayClock=1
UTC=0

[Nextion]
# Port=modem
Port=/dev/ttyAMA0
Brightness=50
DisplayClock=1
UTC=0
IdleBrightness=20

[OLED]
Type=3
Brightness=0
Invert=0
Scroll=1

[LCDproc]
Address=localhost
Port=13666
#LocalPort=13667
DimOnIdle=0
DisplayClock=1
UTC=0
```

## Installazione MMDVMDashboard

Per prima cosa dobbiamo installare e configurare un web server, in questo caso lighttpd, ed sistemare i diritti sulla cartella dei documenti web:

```
sudo apt-get install lighttpd
```

```
sudo usermod -G www-data -a pi
```

```
sudo chown -R www-data:www-data /var/www/html
```

```
sudo chmod -R 775 /var/www/html
```

installiamo anche PHP 5 con le estensioni necessarie:

```
sudo apt-get install php5-common php5-cgi php5
```

```
sudo lighty-enable-mod fastcgi
```

```
sudo lighty-enable-mod fastcgi-php
```

```
sudo service lighttpd force-reload
```

installiamo infine la dashboard:

```
cd /opt
```

```
sudo git clone https://github.com/dg9vh/MMDVMHost-Dashboard.git
```

```
sudo cp -r /opt/MMDVMHost-Dashboard/. /var/www/html/
```

```
cd /var/www/html
```

```
sudo rm index.lighttpd.html
```

```
sudo chown -R www-data:www-data /var/www/html
```

```
sudo chmod -R 775 /var/www/html
```

adesso ci possiamo collegare, con il browser preferito, all'indirizzo dell'hospot (nel mio caso 192.168.2.9) per terminare la configurazione della dashboard tramite il link a setup.php come nelle immagini sotto riportate:

## MMDVM-Dashboard by DG9VH Setup-Process

Please give necessary information below

### MMDVMHost-Configuration

Path to MMDVMHost-logfile	/opt/MMDVMHost/
Path to MMDVM.ini	/opt/MMDVMHost/
MMDVM.ini-filename	MMDVM.ini
Path to MMDVMHost-executable	/opt/MMDVMHost/
Enable extended lookup (show names)	<input checked="" type="checkbox"/>
Show Talker Alias	<input type="checkbox"/>
Path to DMR-ID-Database-File (including filename)	/var/www/html/DMRids.i

# Global Configuration

Timezone	Rome
Locale	en_GB
URL to Logo	http://your-logo
Use networks.php instead of configuration below	<input type="checkbox"/>
URL to DMRplus-Logo	http://your-logo
URL to BrandMeister-Logo	http://your-logo
Refresh page after in seconds	60
Show System Info	<input checked="" type="checkbox"/>
Show Disk Use	<input checked="" type="checkbox"/>
Show Repeater Info	<input checked="" type="checkbox"/>
Show Enabled Modes	<input checked="" type="checkbox"/>
Show Last Heard List of today's	<input checked="" type="checkbox"/>
Show Today's local transmissions	<input checked="" type="checkbox"/>

Show progressbars	<input checked="" type="checkbox"/>
Enable CPU-temperature-warning	<input checked="" type="checkbox"/>
Warning temperature	60
Enable Network-Switching-Function	<input type="checkbox"/>
Enable Reflector-Switching-Function (DMR)	<input type="checkbox"/>
Enable Reflector-Switching-Function (YSF)	<input type="checkbox"/>
Username for switching networks:	username
Password for switching networks:	password
Enable Management-Functions below	<input type="checkbox"/>
Username for view log:	username
Password for view log:	password
Username for halt:	username
Password for halt:	password
Username for reboot:	pi
Password for reboot:	raspberry
Username for restart:	username

Password for restart:	password
Reboot YSFGateway command:	sudo systemctl restart ysf
Reboot MMDVMHost command:	sudo systemctl restart mmdvm
Reboot system command:	sudo reboot
Halt system command:	sudo halt
Show Powerstate (online or battery, wiringpi needed)	<input checked="" type="checkbox"/>
GPIO pin to monitor:	18
State that signalizes online-state:	1
Show link to QRZ.com on Callsigns	<input checked="" type="checkbox"/>
RSSI value	average ▾
Save configuration	

---

adesso dobbiamo scaricare il file con gli ID-DMR:

```
sudo ./cron/updateDMRIDs.sh
```

e schedulare la procedura di aggiornamento automatico con cadenza giornaliera:

```
sudo nano /etc/crontab
```

aggiungere al file la seguente riga (si aggiornerà alle 22:05 di ogni giorno):

```
05 22 * * * root /var/www/html/cron/updateDMRIDs.sh
```

quindi salvare e ricaricare la configurazione:

```
sudo service cron reload
```

non ci rimane, finalmente, che lanciare MMDVMHost:

```
cd /opt/MMDVMHost
```

```
sudo ./MMDVMHost ./MMDVM.ini
```

e verificarne il corretto funzionamento attraverso la dashboard e la radio DMR (dopo averla opportunamente configurata con frequenza uguale a quella impostata nel file **MMDVM.ini**, color code=1 e TS=2 per tutti i TG).

Se tutto è funzionante possiamo rinominare il file **setup.php**

```
cd /var/www/html/
```

```
sudo mv setup.php setup.php.nomore
```

questo per far scomparire l'avvertimento a inizio pagina.

Per una maggiore praticità gestiamo MMDVMHost come un servizio, in modo da avviarlo automaticamente al boot del raspberry:

creiamo il file **mmdvmhost.service**:

```
sudo nano /lib/systemd/system/mmdvmhost.service
```

con il seguente contenuto:

```
[Unit]
Description=MMDVM Host Service
After=syslog.target network.target
[Service]
User=root
WorkingDirectory=/opt/MMDVMHost
ExecStart=/usr/bin/screen -S MMDVMHost -D -m /opt/MMDVMHost/MMDVMHost /opt/MMDVMHost/MMDVM.ini
ExecStop=/usr/bin/screen -S MMDVMHost -X quit
[Install]
WantedBy=multi-user.target
```

ora aggiustiamo i permessi:

```
sudo chmod 755 /lib/systemd/system/mmdvmhost.service
```

e creiamo il link simbolico nella giusta posizione (quello sotto è un unico comando):

```
sudo ln -s /lib/systemd/system/mmdvmhost.service
/etc/systemd/system/mmdvmhost.service
```

creiamo un timer per l'avvio ritardato del servizio:

```
sudo nano /lib/systemd/system/mmdvmhost.timer
```

contenuto del file:

```
[Timer]
OnStartupSec=60
[Install]
WantedBy=multi-user.target
```

soliti permessi da aggiustare:

```
sudo chmod 755 /lib/systemd/system/mmdvmhost.timer
```

e solito link simbolico:

```
sudo ln -s /lib/systemd/system/mmdvmhost.timer
/etc/systemd/system/mmdvmhost.timer
```

attiviamo il servizio:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable mmdvmhost.timer
```

e riavviamo:

```
sudo reboot
```

Se tutto è andato a buon fine, dopo 60 secondi, il nostro hotspot sarà funzionante, potremo monitorarlo dal browser e fare QSO sulla rete DMR brandmeister.

## Installazione DMRGateway

Questo programma deve essere installato solo se vogliamo utilizzare più server DMR o il reflector XLX039 per accedere al modulo B (nazionale) della rete D-Star.

L'installazione è molto simile a quella di MMDVMHost e questi sono i comandi necessari:

```
cd /opt/
```

```
sudo git clone https://github.com/g4klx/DMRGateway
```

```
cd DMRGateway
```

```
sudo make
```

editiamo il file di configurazione **DMRGateway.ini** con il comando:

```
sudo nano DMRGateway.ini
```

La configurazione del file sotto riportato permette di avere la rete brandmeister inalterata, mentre possiamo gestire il collegamento al reflector XLX039 autonomamente, in quanto nell'esperienza fatta non è molto comodo avere la rete D-Star sempre connessa (non è semplice fare QSO sulla rete DMR perché si viene interrotti da un qualsiasi altro QSO sulla rete D-Star) soprattutto se abbiamo la radio settata in "promiscuous mode"

```
[General]
Timeout=10
# RFTIMEOUT=10
# NetTimeout=7
RptAddress=127.0.0.1
RptPort=62032
LocalAddress=127.0.0.1
LocalPort=62031
RuleTrace=1
```

Daemon=0

Debug=0

[Log]

# Logging levels, 0=No logging

DisplayLevel=1

FileLevel=1

FilePath=.

FileRoot=DMRGateway

[Voice]

Enabled=1

Language=it\_IT

Directory=/opt/DMRGateway/Audio

[XLX Network 1]

Enabled=0

Name=XLX039

Address=039.xlxitalia.net

Port=62030

# Local=3351

# Options=

Slot=2

TG=6

Base=60000

Startup=4002

Password=passw0rd

Debug=0

Relink=10

[XLX Network 2]

Enabled=0

Name=XLX000

Address=44.131.4.1

Port=62030

# Local=3351

# Options=

Slot=1

TG=7

Base=74000

Password=passw0rd

Debug=0

# BrandMeister rete primaria --> rimane inalterata

[DMR Network 1]

Enabled=1

Name=BM

Address=brandmeister.digitalham.it

Port=62031

Password=passw0rd

PassAllPC=1

PassAllPC=2

PassAllTG=1

PassAllTG=2

Debug=0

# DMR+

# [DMR Network 2]

# Enabled=0

# Name=DMR+

# Address=[ip address]

```
# Port=55555
# Local=3352
# Password=PASSWORD
# Debug=0

[DMR Network 2]
Enabled=1
Name=XLX039
Address=039.xlxitalia.net
Port=62030
TGRewrite 2,6,2,9,1
TGRewrite 2,9,2,9,1
TGRewrite 2,64000,2,4000,1001
Password=passw0rd
Debug=0
Relink=10
```

Con questa configurazione basta aggiungere al codeplug utilizzato tre TG: TG6, TG64000 e TG64002.

Il TG6 serve per trasmettere/ricevere sulla rete D-Star, il TG6400 per disconnettersi ed il TG64002 per connettersi (questi due TG vanno utilizzati quando siamo sul TG6).

Per utilizzare DMRGateway dobbiamo modificare anche il file MMDVMHost.ini, solo nella parte [DMR Network] come sotto riportato:

```
[DMR Network]
Enable=1
Address=127.0.0.1
Port=62031
Jitter=300
Local=62032
Password=passw0rd
# Options=
Slot1=0
Slot2=1
# ModeHang=3
Debug=0
```

e configurare la parte DMRGateway della dashboard:

```
cd /var/www/html
```

```
sudo cp setup.php.nomore setup.php
```

## DMRGateway-Configuration

Enable DMRGateway	<input checked="" type="checkbox"/>
Path to DMRGateway-logfile	/opt/DMRGateway/
Logfile-prefix	DMRGateway
Path to DMRGateway.ini	/opt/DMRGateway/
Path to DMRGateway-executable	/opt/DMRGateway/
DMRGateway.ini-filename	DMRGateway.ini

quindi, una volta salvata la configurazione:

```
sudo mv setup.php setup.php.nomore
```

possiamo avviare il gateway da riga di comando per testarne il funzionamento:

```
cd /opt/DMRGateway
```

prima dobbiamo arrestare il servizio mmdvmhost:

```
sudo systemctl stop mmdvmhost.service
```

poi avviare il gateway:

```
sudo ./DMRGateway ./DMRGateway.ini &
```

e quindi riavviare il servizio:

```
sudo systemctl start mmdvmhost.service
```

se tutto è ok possiamo gestire anche DMRGateway come servizio

creiamo il file **mmdvmhost.service**:

```
sudo nano /lib/systemd/system/dmrgateway.service
```

con il seguente contenuto:

```
[Unit]
Description=DMRGateway Service
After=syslog.target network.target
[Service]
User=root
WorkingDirectory=/opt/DMRGateway
```

```
ExecStart=/usr/bin/screen -S DMRGateway -D -m /opt/DMRGateway/DMRGateway
/opt/DMRGateway/DMRGateway.ini
ExecStop=/usr/bin/screen -S DMRGateway -X quit
[Install]
WantedBy=multi-user.target
```

ora aggiustiamo i permessi:

```
sudo chmod 755 /lib/systemd/system/dmrgateway.service
```

e creiamo il link simbolico nella giusta posizione (quello sotto è un unico comando):

```
sudo ln -s /lib/systemd/system/dmrgateway.service
/etc/systemd/system/dmrgateway.service
```

creiamo un timer per l'avvio ritardato del servizio:

```
sudo nano /lib/systemd/system/dmrgateway.timer
```

contenuto del file:

```
[Timer]
OnStartupSec=50
[Install]
WantedBy=multi-user.target
```

soliti permessi da aggiustare:

```
sudo chmod 755 /lib/systemd/system/dmrgateway.timer
```

e solito link simbolico:

```
sudo ln -s /lib/systemd/system/dmrgateway.timer
/etc/systemd/system/dmrgateway.timer
```

attiviamo il servizio:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable dmrgateway.timer
```

e riavviamo:

```
sudo reboot
```

al riavvio, dopo circa un minuto, dovrebbe essere tutto funzionante con la rete D-Star non connessa (deve essere connessa manualmente con la procedura vista prima).

Se dobbiamo modificare qualche configurazione o in caso di aggiornamenti, la procedura corretta è la seguente

arrestare il servizio mmdvmhost:

```
sudo systemctl stop mmdvmhost.service
```

arrestare (se necessario) il servizio dmrgateway:

```
sudo systemctl stop dmrgateway.service
```

effettuare le modifiche e riavviare il servizio dmrgateway:

```
sudo systemctl start dmrgateway.service
```

riavviare il servizio mmdvmhost:

```
sudo systemctl start mmdvmhost.service
```

## Configurazione rete Wireless

Se anziché la rete cablata vogliamo usare l'interfaccia di rete wireless presente sul Raspberry pi 3 basta creare il file `/etc/wpa_supplicant/wpa_supplicant.conf` con il solito comando:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

con il seguente contenuto:

```
country=IT
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
ssid="SSID_RETE"
psk="password_rete"
key_mgmt=WPA-PSK
}
```

eventualmente possiamo inserire più reti ripetendo per ogni rete la parte “network” (vedi esempio sotto) aggiungendo a tutte l'id\_str e, se vogliamo, la priorità (nel caso che ce ne sia più di una raggiungibile contemporaneamente verrà collegata quella con priorità più alta)

```
network={
ssid="SSID_RETE"
psk="password_rete"
key_mgmt=WPA-PSK
priority=1
id_str="stringa_di_identificazione_rete"
}
```

nel caso di collegamento a reti nascoste dobbiamo aggiungere anche l'opzione:

```
scan_ssid=1
```

nel caso invece di reti aperte dobbiamo omettere l'opzione **psk** ed assegnare NONE a **key\_mgmt**.

# Bibliografia

Questo documento è stato redatto attingendo alla documentazione dei software installati e da altri documenti presenti in rete, queste le fonti principali:

- Gruppo telegram “DMR in Toscana” (vari contributi dei membri)
- [Chris Andrist, KC7WSU \(DMR-UTAH\) Preparing Raspberry Pi](#)
- [Filip, ON3FV](#)
- [ThüringenLink](#)
- [Mario Radtke, DC7JZB](#)
- [F5UII](#)