

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 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 comando¹, senza l'installazione di nessuna interfaccia grafica.

Buon divertimento !!

Materiale impiegato

Hardware²

- 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

Software³

- [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

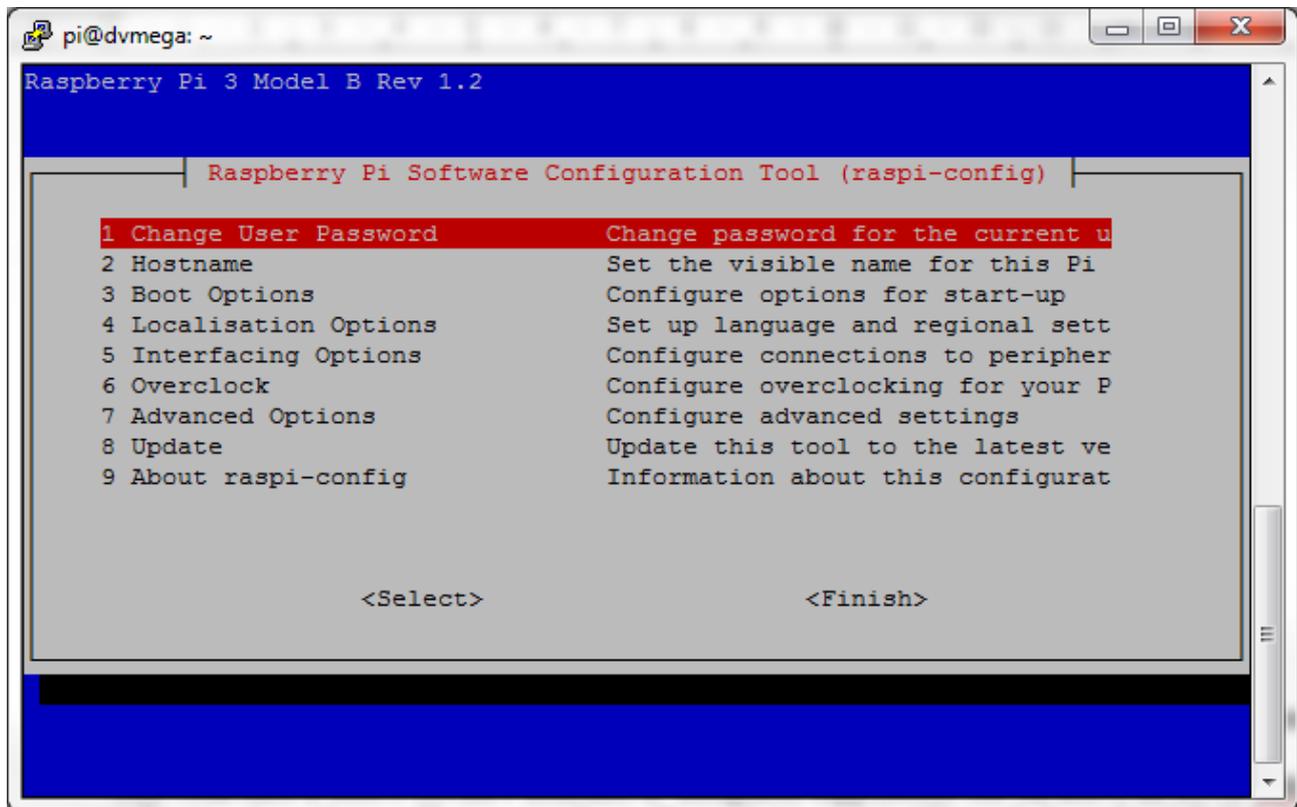
² Tutte le prove sono state eseguite con una radio TYT MD380

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

Una volta fatto il boot e trovato l'indirizzo IP assegnato⁴, 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:



- Hostname⁵
- 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=xxxxxx¹¹
URL=https://qrz.com/db/xxxxxx¹²

[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 php-common php-cgi php
```

```
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.r

Global Configuration

Timezone	Rome ▾
Locale	en_GB ▾
URL to Logo	<input type="text" value="http://your-logo"/>
Use networks.php instead of configuration below	<input type="checkbox"/>
URL to DMRplus-Logo	<input type="text" value="http://your-logo"/>
URL to BrandMeister-Logo	<input type="text" value="http://your-logo"/>
Refresh page after in seconds	<input type="text" value="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:	raspberrypi
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 DMRGateway¹³

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
```

¹³ Prima di installare questo software leggere il paragrafo "Commutazione rete via Dashboard o TG"

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**.

Ottimizzazioni

Una volta testato l'hotspot, se decidiamo di utilizzarlo definitivamente, ci sono alcune ottimizzazioni da fare per renderne più agevole la gestione (eventuali aggiornamenti, ecc.) e più robusto il sistema.

Per prima cosa copiamo eseguibili e file di configurazione in cartelle diverse da quelle in cui sono stati compilati.

Per gli eseguibili:

```
cd /usr/local/sbin/  
  
sudo cp /opt/MMDVMHost/MMDVMHost .  
  
sudo cp /opt/DMRGateway/DMRGateway .
```

Per i files di configurazione:

```
cd /usr/local/etc/  
  
sudo cp /opt/MMDVMHost/MMDVM.ini .  
  
sudo cp /opt/DMRGateway/DMRGateway.ini .
```

Quindi dobbiamo editare i servizi creati prima sostituendo i percorsi con quelli nuovi (vedi righe evidenziate sotto):

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

nuovo file:

```
[Unit]  
Description=MMDVM Host Service  
After=syslog.target network.target  
[Service]  
User=root  
WorkingDirectory=/opt/MMDVMHost  
ExecStart=/usr/bin/screen -S MMDVMHost -D -m /usr/local/sbin/MMDVMHost  
/usr/local/etc/MMDVM.ini  
ExecStop=/usr/bin/screen -S MMDVMHost -X quit  
Restart=always  
RestartSec=10  
[Install]  
WantedBy=multi-user.target
```

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

nuovo file:

```
[Unit]  
Description=DMRGateway Service  
After=syslog.target network.target
```

```
[Service]
User=root
WorkingDirectory=/opt/DMRGateway
ExecStart=/usr/bin/screen -S DMRGateway -D -m /usr/local/sbin/DMRGateway
/usr/local/etc/DMRGateway.ini
ExecStop=/usr/bin/screen -S DMRGateway -X quit
[Install]
WantedBy=multi-user.target
```

Come seconda ottimizzazione scriviamo i log in RAM anziché sulla scheda SD, questo ha due importanti benefici: si limitano le scritture sulla SD in modo da aumentarne la durata e si velocizza la dashboard, come contropartita ogni volta che riavviamo il raspberry si perderanno i log, ma visto che l'uso è principalmente quello di visualizzarli nella dashboard, è importante avere solo quelli attuali:

Creiamo la cartella dove monteremo il RAMDisk:

```
cd /tmp/

mkdir ram

chmod 777 ram
```

Creiamo il disco di 50MB

```
sudo mount -t tmpfs -o size=50m tmpfs /tmp/ram
```

ed automatizziamo il tutto:

```
sudo vi /etc/fstab
```

riga da aggiungere in fondo:

```
tmpfs /tmp/ram tmpfs defaults,noatime,size=50m 0 0
```

a questo punto dobbiamo riaggiornare la configurazione dei servizi:

```
sudo systemctl daemon-reload
```

arrestarli e riavviarli:

```
sudo systemctl stop mmdvmhost.service

sudo systemctl stop dmrgateway.service

sudo systemctl start dmrgateway.service

sudo systemctl start mmdvmhost.service
```

e modificare la configurazione della dashboard inserendo i percorsi corretti per i files spostati:

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

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

modifichiamo i percorsi come quelli delle immagini:

MMDVMHost-Configuration

Path to MMDVMHost-logfile	/tmp/ram/
Path to MMDVM.ini	/usr/local/etc
MMDVM.ini-filename	MMDVM.ini
Path to MMDVMHost-executable	/usr/local/sbin/
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.1

DMRGateway-Configuration

Enable DMRGateway	<input checked="" type="checkbox"/>
Path to DMRGateway-logfile	/tmp/ram/
Logfile-prefix	DMRGateway
Path to DMRGateway.ini	/usr/local/etc/
Path to DMRGateway-executable	/usr/local/sbin/
DMRGateway.ini-filename	DMRGateway.ini

eliminiamo l'avvertimento rinominando il file:

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

Infine creiamo una procedura automatica per eliminare i log più vecchi di una settimana, per fare questo basta aggiungere una riga in fondo al file `/etc/crontab` come sotto riportato:

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

riga da aggiungere in fondo (in questo esempio i log più vecchi di 7 giorni verranno cancellati alle ore 03:12):

```
12 3 * * * root find /tmp/ram/* -mtime +7 -exec rm {} \;
```

e quindi ricaricare la nuova configurazione:

```
sudo service cron reload
```

Con queste ottimizzazioni l’hotspot è pronto per l’utilizzo in modo continuato, senza doversi preoccupare del riempimento della SD dovuto ai log scritti giornalmente.

Commutazione rete via Dashboard o TG

La configurazione proposta ha un grosso svantaggio che è quello di rimanere “intrappolati” in altri QSO rispetto al nostro e questo perché utilizzando DMRGateway ed un solo slot tutte le reti sono “sovrapposte”, quindi se sto facendo un QSO sulla rete D-STAR e tra un passaggio e l’altro c’è un QSO (magari con passaggi lunghi e/o molto “stretti”), non possiamo continuare il nostro.

Per ovviare a questo inconveniente è stato scelto di fare un passo indietro e di ritornare al solo utilizzo di MMDVMHost con file di configurazione multipli e di utilizzare la funzione di “Network Switching” presente nella Dashboard, per poi ampliarne il funzionamento e poterla utilizzare anche via radio tenendo comunque allineata la scelta fatta indipendentemente dal metodo utilizzato.

Per prima cosa dobbiamo impostare correttamente il database delle configurazioni che vogliamo commutare, per fare questo basta editare il file `/var/www/html/config/networks.php` nel quale è stato aggiunto un campo dal nome **TG** che dobbiamo valorizzare con il TG impiegato per caricare via radio quella configurazione, eccone un esempio con i relativi files di configurazione (solo la parte relativa al DMR):

```
/var/www/html/config/networks.php
```

```
<?php
$networks_json = '{
    "DMRPLUS":{
        "label":"Multimode DMRplus",
        "ini":"DMRPLUS",
        "logo":"http://db0lm.de/wp-content/uploads/2015/10/DMR-Plus-
300x109.png",
        "TG": "999991"
    },
    "BRANDMEISTER":{
        "label":"Multimode BrandMeister",
        "ini":"BRANDMEISTER",
        "logo":"https://s3.amazonaws.com/files.qrz.com/a/pd1ra/dmr_brandmeister.jp
g",
        "TG": "999992"
    },
    "DSTAR":{
        "label":"Singlemode D-STAR",
        "ini":"DSTAR",
        "logo":"http://www.areg.org.au/wp-
content/uploads/2015/10/wncdstar_logo.png",
```

```

        "TG": "999993"
    }
};

$networks = json_decode($networks_json, true);
?>

```

/usr/local/etc/DMRPLUS.ini

```

[DMR Network]
Enable=1
Address=dmmaster.grupporadiofirenze.net
Port=55555
Jitter=300
Password=PASSWORD
Options=StartRef=4255;RelinkTime=15;UserLink=1;
Slot1=0
Slot2=1
Debug=2

```

/usr/local/etc/BRANDMEISTER.ini

```

[DMR Network]
Enable=1
Address=brandmeister.digitalham.it
Port=62031
Jitter=300
Password=passwd
Slot1=0
Slot2=1
Debug=2

```

/usr/local/etc/DSTAR.ini

```

[DMR Network]
Enable=1
Address=039.xlxitalia.net
Port=62030
Jitter=300
Password=passwd
Options=StartRef=4002;RelinkTime=15;UserLink=1;
Slot1=0
Slot2=1
Debug=2

```

Quindi, per poter comutare rete tramite i pulsanti presenti sulla dashboard bisogna concedere i permessi di **root**, tramite `sudo`, all'utente **www-data** (utente che esegue il server web) aggiungendo questa riga, impiegando `visudo`, al file **etc/sudoers**:

```
sudo visudo
```

aggiungere, in fondo, la seguente riga:

```
www-data ALL=(ALL) NOPASSWD: ALL14
```

14 Con questa riga si concedono tutti i permessi di root all'utente **www-data**, per il nostro scopo potremmo usare una politica più restrittiva facendo eseguire solo i comandi strettamente necessari. Supponendo però che il server web sia solo accessibile tramite intranet possiamo, per semplicità, concederli tutti.

Infine dobbiamo correttamente configurare la dashboard per l'uso della funzione di network switching, come da immagini sotto (checkbox indicati dalla freccia):

Global Configuration

Timezone	Rome
Locale	en_GB
URL to Logo	http://your-logo
Use networks.php instead of configuration below	<input checked="" type="checkbox"/>
Warning temperature	60
Enable Network-Switching-Fuction	<input checked="" type="checkbox"/>
Enable DMRplus Switching-Fuction (DMR)	<input type="checkbox"/>

A questo punto possiamo verificarne il corretto funzionamento tramite i pulsanti come da immagine:



controllando, tramite il comando:

```
sudo tail -F /tmp/ram/MMDVM-$(date -u +%F).log
```

il corretto riavvio di MMDVMHost, il successivo collegamento al server selezionato ed il corretto aggiornamento della dashboard.

Adesso possiamo configurare anche la parte di commutazione rete via radio, per fare questo è stato creato uno script php che esegue il codice che è associato alla pressione dei tasti di commutazione della configurazione:

```
/var/www/html/local/switch_network.php
```

```
<?php
```

```

// script da eseguire per cambio configurazione
// via chiamata a specifico TG
// da richiamare come root con argomento il TG

include "../config/config.php";
include("../config/networks.php");
include "../include/tools.php";
include "../include/functions.php";
include "../include/init.php";

if ($argc > 1) {
$netname = $argv[1];
$key = recursive_array_search($netname,$networks);
if (strlen($key) > 0) {
$network = $networks[$key];
setDMRNetwork($network['label']);
exec("sudo cp ".MMDVMINIPATH."/".$network['ini'].".ini
".MMDVMINIPATH."/".MMDVMINIFILENAME);
exec(REBOOTMMDVM);
}
}
?>

```

ed è stato modificato uno script già presente nei sorgenti di MMDVMHost (linux/tg_shutdown.sh) come segue (parti in grassetto), sono state comunque preservate le funzioni originali (shutdown e reboot):

/usr/local/sbin/tg_switch.sh

```

#!/bin/bash

# tg_shutdown.sh - Automated shutdown and reboot on receipt of talkgroup
# Copyright (C) 2017 Tony Corbett, G0WFV and Stuart Scott, VK6LS
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USAp

# exit status ...
# 1 script not run as root
# 2 ini file doesn't exist

### CONFIG VARIABLES ###

sysopCallsign=XXXXXX15
iniFile=/usr/local/etc/MMDVM.ini
startswTG=999991
stopswTG=999993
shutdownTG=999999
rebootTG=888888
allowShutdown=0
allowReboot=0

```

15 Inserire il nominativo.

```

switchTGscript='/var/www/html/local/switch_network.php'

### DON'T EDIT BELOW HERE ###

# exit if we're not root ...
[[ $EUID -ne 0 ]] && exit 1

# exit if we can't find the ini file
[[ -f $iniFile ]] || exit 2

# process the inifile and convert to variables ...
#
# [Foo]
# Bar=Baz
#
# ... becomes the variable $FooBar with the value "Baz"
# (NOTE: spaces in section headers are replaced by underscores)
#
# a real life example ...
#
# [General]
# Callsign=G0WfV
#
# ... becomes the variable $GeneralCallsign with the value "G0WfV"

foo=$(
  cat $iniFile | while read line
  do
    if echo $line | grep '^#.*$' >/dev/null # comment line
    then
      # Ignore!
      continue
    elif echo $line | grep '^$' >/dev/null # blank line
    then
      # Ignore!
      continue
    elif echo $line | grep '^\[.*\]$' >/dev/null # [Section Header]
    then
      iniSection=$(echo $line | sed 's/^\[.*\]$/\1/' | sed 's/[ -]//g')
    elif echo $line | grep '^.*=.*$' >/dev/null # Key=Value pair
    then
      iniKey=$(echo $line | sed 's/\(.*\)=.*/\1/')
      iniValue=$(echo $line | sed 's/.*=\(.*\)/\1/')
      echo $iniSection$iniKey="$iniValue" # print the result in a
specific format
    else # hopefully we'll never get here, but you never know!
      continue
    fi
  done
)
eval $foo

# fix filepath if it doesn't end with a /
[[ "$(echo ${LogFilePath: -1})" != "/" ]] && LogFilePath="$LogFilePath/"

currentDate=foo # dummy current date variable to kick off the 1st tail!
shuttingDown=0

while true # main outer loop (run this forever!)
do
  checkDate=$(date -u +%Y-%m-%d)
  if [ "$checkDate" != "$currentDate" ]
  then
    kill $tailPID 2>/dev/null
    currentDate=$checkDate
    logFile=$LogFilePath$LogFileRoot-$currentDate.log

    tail -n 0 -F $logFile | while read line # inner loop to tail the logfile
    do

```

```

# only react to sysop callsign ...
foo=$(echo $line | grep "received RF voice header from $sysopCallsign
to TG")

if [ $? = 0 ]
then
    TG=$(echo $line | sed "s/.*TG\(.*\)$/\1/g")

    if [ $TG -eq $shutdownTG ] && [ $shuttingDown -eq 0 ] &&
[ $allowShutdown -eq 1 ]
    then
        # shutdown in 1 minute ...
        shutdown -h +1 >/dev/null 2>&1 && shuttingDown=1
    elif [ $TG -eq $rebootTG ] && [ $shuttingDown -eq 0 ] &&
[ $allowReboot -eq 1 ]
    then
        # reboot in 1 minute ...
        shutdown -r +1 >/dev/null 2>&1 && shuttingDown=1
    elif [ $shuttingDown -eq 1 ]
    then
        # cancel shutdown or reboot if sysop tx any TG in 1 min
        grace period ...
        shutdown -c && shuttingDown=0
    elif [ $TG -ge $startswTG ] && [ $TG -le $stopswTG ]
    then
        # exec switch TG script
        current_dir=$PWD;cd $(dirname $switchTGscript);php $switchTGscript $TG;cd
$current_dir;
        fi
    fi
done & 2>/dev/null # inner loop is run in background so we can periodically
check if the date's changed
tailPID=$((#! - 1)) # save the PID of the inner loop so we can kill it when
the date rolls over
fi
sleep 1 # check every second for date rollover (reduces cpu load)
done

```

Infin non rimane che creare un servizio nella solita maniera già vista, per prima cosa creiamo il file:

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

con il seguente contenuto:

```

[Unit]
Description=TG switch Service
After=syslog.target network.target

[Service]
User=root
WorkingDirectory=/tmp/ram
ExecStart=/usr/bin/screen -S tgswitch -D -m /usr/local/sbin/tg_switch.sh
ExecStop=/usr/bin/screen -S tgswitch -X quit
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target

```

ora aggiustiamo i permessi:

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

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

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

creiamo un timer per l'avvio ritardato del servizio:

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

contenuto del file:

```
[Timer]
OnStartupSec=70
[Install]
WantedBy=multi-user.target
```

soliti permessi da aggiustare:

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

e solito link simbolico:

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

attiviamo il servizio:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable tgswitch.timer
```

e riavviamo:

```
sudo reboot
```

Il nostro hotspot è adesso pronto per l'uso, ricordando che possiamo cambiare rete attraverso i pulsanti presenti sulla dashboard oppure dando un colpo di PTT sui TG che abbiamo impostato nel file di configurazione.

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](#)