

Pi-Star für lokale Benutzung einrichten

In diesem Beispiel verwende ich einen Raspberry Pi 3 (wg. Ethernetanschluss) mit einem MMDVM Hat Klon.

Lade das aktuelle Pi-Star Image herunter und schreibe es auf die Karte.

Das Image findest du hier: <http://pistar.uk>

Default username: **pi-star**

Default password: **raspberrypi**

1. Pi-Star für Internetbetrieb einrichten

Konfiguriere die Bereiche nacheinander und bestätige „Änderung Speichern“ nach jedem Bereich.

Hier ist meine Beispielkonfiguration:

Pi-Star:3.4.16 / Dashboard: 20180806

Pi-Star Digital Voice - Configuration

Dashboard | Admin | Expert | Power | Update | Backup/Restore | Factory Reset

Gateway Hardware Information

Hostname	Kernel	Platform	CPU Load	CPU Temp
pi-star	4.9.35-v7+	Pi 3 Model B (1GB) - Sony, UK	0.89 / 0.5 / 0.21	41.9°C / 107.4°F

Control Software

Setting	Value
Controller Software:	<input type="radio"/> DStarRepeater <input checked="" type="radio"/> MMDVMHost (DV-Mega Minimum Firmware 3.07 Required)
Controller Mode:	<input checked="" type="radio"/> Simplex Node <input type="radio"/> Duplex Repeater (or Half-Duplex on Hotspots)

MMDVMHost Configuration

Setting	Value
DMR Mode:	<input checked="" type="checkbox"/> RF Hangtime: 20 Net Hangtime: 20
D-Star Mode:	<input type="checkbox"/> RF Hangtime: 20 Net Hangtime: 20
YSF Mode:	<input type="checkbox"/> RF Hangtime: 20 Net Hangtime: 20
P25 Mode:	<input type="checkbox"/> RF Hangtime: 20 Net Hangtime: 20
NXDN Mode:	<input type="checkbox"/> RF Hangtime: 20 Net Hangtime: 20
YSF2DMR:	<input type="checkbox"/>
YSF2NXDN:	<input type="checkbox"/>
YSF2P25:	<input type="checkbox"/>
DMR2YSF:	<input type="checkbox"/> Uses 7 prefix on DMRGateway
DMR2NXDN:	<input type="checkbox"/> Uses 7 prefix on DMRGateway
POCSAG:	<input type="checkbox"/> POCSAG Paging Features
MMDVM Display Type:	OLED ▼ Port: /dev/ttyAMA0 ▼ Nextion Layout: G4KLX ▼

General Configuration

Setting	Value
Hostname:	da0not-dmr <small>Do not add suffixes such as .local</small>
Node Callsign:	DA0NOT
CCS7/DMR ID:	2627576
Radio Frequency:	433.625.000 MHz
Latitude:	48.99 degrees (positive value for North, negative for South)
Longitude:	8.34 degrees (positive value for East, negative for West)
Town:	Karlsruhe, JN48EX
Country:	Germany
URL:	http://www.qsl.net/db/da0not <input type="radio"/> Auto <input checked="" type="radio"/> Manual
Radio/Modem Type:	MMDVM_HS_Hat (DB9MAT & DF2ET) for Pi (GPIO)
Node Type:	<input type="radio"/> Private <input checked="" type="radio"/> Public
System Time Zone:	Europe/Berlin
Dashboard Language:	german_de

Apply Changes

DMR Configuration

Setting	Value
DMR Master:	BM_Germany_2622
BrandMeister Network:	Repeater Information Edit Repeater (BrandMeister Selfcare)
DMR Colour Code:	1
DMR EmbeddedLCOnly:	<input type="checkbox"/>
DMR DumpTADData:	<input checked="" type="checkbox"/>

Apply Changes

Im Bereich „Allgemeine Einstellungen“ Node Typ = Public“ (jede ID kann den Hotspot benutzen).

Viel Spaß. Der Pi-Star ist jetzt für normalen Internetbetrieb konfiguriert.

Es gibt im Netz genug Tutorials und Videos zu dem Thema...

2. Lokales DMR Netzwerk einrichten

Installation von HBLink:

HBLink kann auf einem separaten RasPi oder Linux PC installiert werden. Der Versuch HBLink auf den gleichen RasPi zu installieren wie den Hotspot hat zwar funktioniert, jedoch konnten externe Hotspots den Server nicht erreichen. Anscheinend hängt das mit der eingebauten Firewall von Pi-Star zusammen. Aus einer nativen MMDVM Installation sollte es keine Probleme geben.

Als erstes wird der Name des Servers mit

```
sudo raspi-config geändert (hier: DMRMaster).
```

Laden der Updates:

```
sudo apt update  
sudo apt upgrade
```

Installieren von git:

```
sudo apt install git
```

Runterladen der Pakete:

```
git clone https://github.com/n0mjs710/dmr_utils  
git clone https://github.com/n0mjs710/HBLink
```

Automatisches Installieren der dmr_utils:

```
cd dmr_utils  
sudo ./install.sh
```

Wechseln in den HBLink Ordner:

```
cd ..  
cd HBLink/
```

Anpassen der Konfiguration:

```
cp hblink-SAMPLE.cfg hblink.cfg  
nano hblink.cfg
```

Folgende Änderungen sind notwendig:

```
[LOGGER]  
...  
LOG_LEVEL: INFO  
...  
  
[MASTER-1]  
MODE: MASTER  
ENABLED: True  
REPEAT: True  
EXPORT_AMBE: False  
IP:  
PORT: 62031
```

```
PASSPHRASE: s3cr37w0rd
GROUP_HANGTIME: 5
...

[REPEATER-1]
...
ENABLED: False
```

Die Elemente „LOG_LEVEL“, der „MASTER-1 port“ und das Deaktivieren von „REPEATER-1“ sind relevant.

Pi-Star hat seine eigene Firewall und blockt standardmäßig viele Ports. Die Verwendung von Port 62031 (Brandmeister) löst das Problem.

Speichern und Beenden des Editors (**CTRL + X, Y, ENTER**)

3. HBLink beim Boot automatisch ausführen

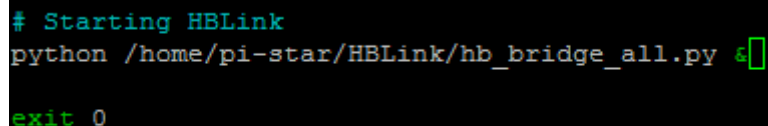
Anpassen der Autostartdatei:

```
sudo nano /etc/rc.local
```

Diese Zeilen müssen vor „exit 0:“ eingefügt werden.

```
# Starting HBLink
python /home/pi-star/HBLink/hb_bridge_all.py &
```

So sollte das ganze dann aussehen:



```
# Starting HBLink
python /home/pi-star/HBLink/hb_bridge_all.py &
exit 0
```

Speichern und Beenden nicht vergessen...

Speichern und Beenden des Editors (**CTRL + X, Y, ENTER**)

HBLink RasPi neu starten um die Konfiguration zu aktivieren

```
sudo reboot
```

4. AREDN DHCP Address Reservierung erstellen

The screenshot shows the AREDN web interface for a node named 'DL4FLY-052-MTRB'. The 'Port Forwarding, DHCP, and Services' tab is active. The interface includes navigation links for Node Status, Basic Setup, Port Forwarding, DHCP, and Services, Tunnel Server, Tunnel Client, and Administration. Below these are buttons for Help, Save Changes, Reset Values, and Refresh.

DHCP Address Reservations

Hostname	IP Address	MAC Address	
da0not-dmr	10.97.253.57	[Redacted]	Del
DMRMaster	10.97.253.55	[Redacted]	Del
	- IP Address -		Add

Advertised Services

Name	Link	URL	
MeshCha	<input checked="" type="checkbox"/>	http://DL4FLY-052-MTRB:8080/meshchat	Del
	<input type="checkbox"/>	::DL4FLY-052-MTRB: /	Add

Current DHCP Leases

da0not-dmr	10.97.253.57	[Redacted]	Add
DL4FLY-CFF9	10.97.253.62	[Redacted]	Add
DMRMaster	10.97.253.55	[Redacted]	Add

Port Forwarding

Interface	Type	Outside Port	LAN IP	LAN Port	
WAN	TCP		- IP Address -		Add

HBLink sollte nun mit seinem Rechnernamen auf der Mesh-Status-Seite erscheinen:

Local Hosts	Services
DL4FLY-052-MTRB.local.mesh (wan)	MeshChat-A12
● DMRMaster.local.mesh	

5. Pi-Star für Lokalbetrieb anpassen

In Pi-Star gehe zu **Konfiguration > Expert > MMDVMHost**

Weiter unten finden sich im Bereich „DMR Network“ die relevanten Einstellungen. Hier werden die Parameter aus HBLink eingetragen. Es können beide Zeitschlitze verwendet werden.

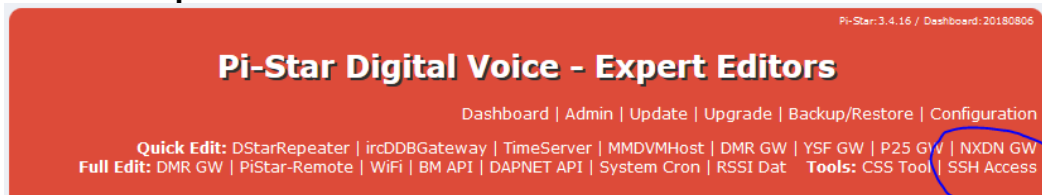
DMR Network	
Enable	1
Address	DMRMaster.local.mesh
Port	62031
Jitter	360
Password	s3cr37w0rd
Slot1	1
Slot2	1
Debug	0
ModeHang	20

System Fusion Network

Test der Verbindung:

Über Putty oder die Pi-Star GUI wird eine SSH Konsole geöffnet

Konfiguration > Expert > SSH Access



Über das Log-File kann geschaut werden ob die Anmeldung am Master erfolgreich war:

```
tail -f /var/log/pi-star/MMDVM-2018xxxx.log
```

Pi-Star erzeugt für jeden Tag ein neues Log-File.

Das Log sollte ähnlichen Meldungstext bringen:

```
pi-star@da0not-dmr(ro):~$ tail -f /var/log/pi-star/MMDVM-2018-10-07.log
I: 2018-10-07 10:34:25.846 Call Hang: 3s
I: 2018-10-07 10:34:25.846 TX Hang: 4s
I: 2018-10-07 10:34:25.846 Mode Hang: 20s
I: 2018-10-07 10:34:26.030 Interfaces Info
I: 2018-10-07 10:34:26.031 IPv4: lo:127.0.0.1
I: 2018-10-07 10:34:26.031 IPv4: eth0:192.168.1.110
I: 2018-10-07 10:34:26.031 Default interface is : eth0
I: 2018-10-07 10:34:26.032 IP to show: eth0:192.168.1.110
M: 2018-10-07 10:34:26.032 MMDVMHost-20180802_Pi-Star is running
M: 2018-10-07 10:34:35.870 DMR, Logged into the master successfully
```

Viel Spaß, 73 Timm DL4FLY