Nutzung mit MMDVM und BlueStack micro+

Stand: 2025/10/15 22:13

Autor: Kim Hübel - DG9VH, dg9vh@darc.de

Inhaltsverzeichnis

Nutzung mit MMDVM und BlueStack micro+ 5
Voraussetzungen 5
Vorbereitung des DVMEGA 5
Vorbereitung des Raspberry Pi 5
Installation und Konfiguration von MMDVMHost 5
Startskript für automatischen Start beim Hochfahren des Raspberry Pi 8

Nutzung mit MMDVM und BlueStack micro+

Voraussetzungen

Zur Nutzung des DVMEGA mit MMDVM und BlueStack micro+ benötigt man

- ein DVMEGA RPI RADIO Modul
- ein BlueStack micro+
- einen Raspberry Pi mit Internet-Zugang
- die Software MMDVMHost von G4KLX

Vorbereitung des DVMEGA

Bevor das DVMEGA Modul zusammen mit der blue Spot App für DMR benutzt werden kann, ist zunächst ein Firmware-Update vorzunehmen. Hierzu montiert man das Modul auf das BlueStack Modul, rastet den auf dem BlueStack Modul befindlichen Schalter ein, schließt beides per USB an einen PC an und führt die xloader Software aus um das Firmware Update einzuspielen.

Hierbei ist wichtig als Typ des Gerätes “UNO“ auszuwählen.

Nachdem das Firmware-Update vorgenommen wurde trennt man wieder alles vom USB ab, schließt alles per USB an den Raspberry Pi an und startet die MMDVMHost-Software, nachdem diese konfiguriert wurde.

Vorbereitung des Raspberry Pi

Um die MMDVMHost-Software erfolgreich herunterzuladen und zu kompilieren, sollte man folgende Befehlssequenzen ausfühen:

```
sudo apt-get update
sudo apt-get install git build-essential gcc
```

Installation und Konfiguration von MMDVMHost

Zunächst erzeugt man sich eine lokale Kopie der Quelldateien (evtl. in einem eigenen Verzeichnis /home/pi/src/) mit dem Befehl

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

anschließend wechselt man in das erzeugte Verzeichnis

```
cd MMDVMHost
```

und stößt den Compiliervorgang an

```
make
```

Nach dem Kompilieren legt man sich mit

```
sudo mkdir /opt/mmdvm
```

ein entsprechendes Verzeichnis an, welches den MMDVMHost zukünftig beherbergt.

Im Anschluss kopiert man die beiden Dateien „MMDVMHost“ und „MMDVM.ini“ nach „/opt/mmdvm/“ und editiert die MMDVM.ini-Datei etwa nach folgendem Muster:

```
[General]
Callsign=DG9VH
Timeout=180
Duplex=0
ModeHang=10
Display=None

[Info]
RXFrequency=433612500
TXFrequency=433612500
Power=0.01
Latitude=0.0
Longitude=0.0
Height=0
Location=Voelklingen, JN39kf
Description=Multi-Mode Repeater
URL=www.dg9vh.de

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

[Modem]
# Port=/dev/ttyACM0
#Port=\\.\COM3
Port=/dev/ttyUSB0
TXInvert=1
RXInvert=0
PTTInvert=1
TXDelay=100
DMRDelay=0
RXLevel=50
TXLevel=50
OscOffset=50
Debug=1
```

```
[D-Star]
Enable=0
Module=C

[DMR]
Enable=1
Beacons=1
Id=2625094
ColorCode=1

[System Fusion]
Enable=0
Parrot=1

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

[DMR Network]
Enable=1
Address=master.up4dar.de
Port=62031
Password=passw0rd
Slot1=0
Slot2=1
Debug=1

[System Fusion Network]
Enable=0
Address=44.131.4.1
Port=32768
Debug=1

[TFT Serial]
#Port=/dev/ttyAMA0
#Brightness=50

[HD44780]
#Rows=2
#Columns=16
```

Anschließend kann man die Software mit

```
sudo /opt/mmdvm/MMDVMHost /opt/mmdvm/MMDVM.ini
```

starten.

Startskript für automatischen Start beim Hochfahren des Raspberry Pi

An dieser Stelle wird es leider jetzt etwas unsauber, da man zunächst mittels

```
sudo chmod +s /opt/mmdvm/MMDVMHost
```

dem MMDVMHost ein „sticky bit“ verpassen muss, damit es die Logs schreiben kann.

Hier das Startscript:

```
#!/bin/bash
### BEGIN INIT INFO
#
# Provides:             MMDVMHost
# Required-Start:       $remote_fs
# Required-Stop:        $remote_fs
# Default-Start:        2 3 4 5
# Default-Stop:         0 1 6
# Short-Description:    MMDVMHost initscript

#
### END INIT INFO
## Fill in name of program here.
PROG="MMDVMHost"
PROG_PATH="/opt/mmdvm/"
PROG_ARGS="/opt/mmdvm/MMDVM.ini"
PIDFILE="/var/run/mmdvm.pid"
USER="pi"

start() {
    if [ -e $PIDFILE ]; then
        ## Program is running, exit with error.
        echo "Error! $PROG is currently running!" 1>&2
        exit 1
    else
        ## Change from /dev/null to something like /var/log/$PROG if you
want to save output.
        cd $PROG_PATH
        sleep 30
        sudo -u $USER /usr/bin/stdbuf -i0 -o0 -e0 ./$PROG $PROG_ARGS 2>&1 >>
/dev/null &
        echo "$PROG started"
        touch $PIDFILE
    fi
}

stop() {
    if [ -e $PIDFILE ]; then
```

```
        ## Program is running, so stop it
       echo "$PROG is running"
       rm -f $PIDFILE
       killall $PROG
       echo "$PROG stopped"
    else
        ## Program is not running, exit with error.
        echo "Error! $PROG not started!" 1>&2
        exit 1
    fi
}

## Check to see if we are running as root first.
## Found at http://www.cyberciti.biz/tips/shell-root-user-check-script.html
if [ "$(id -u)" != "0" ]; then
    echo "This script must be run as root" 1>&2
    exit 1
fi

case "$1" in
    start)
        start
        exit 0
    ;;
    stop)
        stop
        exit 0
    ;;
    reload|restart|force-reload)
        stop
        sleep 2
        start
        exit 0
    ;;
    **)
        echo "Usage: $0 {start|stop|reload}" 1>&2
        exit 1
    ;;
esac
exit 0
### END
```

Dieses Script als /etc/init.d/mmdvm.sh abspeichern und mit

```
sudo chkconfig mmdvm.sh on
```

aktivieren.

From:
https://wiki.dg9vh.de/ - **DG9VH Wiki**

Permanent link:
https://wiki.dg9vh.de/dmr:dvmega:nutzung_mit_mmdvm_und_bluestack_micro?rev=1458917945

Last update: **2016/03/25 15:59**

Falls nicht anders bezeichnet, ist der Inhalt dieses Dokuments unter der folgenden Lizenz veröffentlicht: CC Attribution-Share Alike 4.0 International