

lindmx - DMX für Linux

florian.schmitt@qikgrp.com

8. Juli 2013

Inhaltsverzeichnis

1	Einführung	2
1.1	Vorwort: Was ist LINDMX?	2
1.2	Installation: Debian, Ubuntu, Raspbian	2
1.3	Installation: openSUSE	2
1.4	Deinstallation	2
1.5	Inbetriebnahme: DMX4ALL Micro USB	3
1.6	Inbetriebnahme: DMX4ALL LAN	3
1.7	Benutzung: Starten des Servers	3
1.8	Benutzung: Steuerung	4
1.9	Konfigurationsdatei: /etc/lindmxd.conf	5
1.10	Server beim Systemstart starten	5
1.11	Sicherheit	5
1.12	DMX in Shell-Skripten	5
1.13	DMX mit Cronjobs	6
2	dmxprog: Show Programmer	7
2.1	Einführung	7
2.1.1	Begriffe	7
2.1.2	Allgemeines	7
2.1.3	Konzepte der Wiedergabe	8
2.1.4	Programmieren von Chasern	8
2.1.5	Einstellungen von Chasern	9
2.1.6	Bearbeiten von aufgenommenen Szenen	9
2.2	Die Kommandozeile	10
2.2.1	Allgemeine Befehle	10
2.2.2	Befehle zur Steuerung der Kanäle	10
2.2.3	Befehle zur Steuerung der Chaser	11
2.2.4	Befehle zur Einrichten von MIDI-Fernsteuerung	11

1 Einführung

1.1 Vorwort: Was ist LINDMX?

LINDMX ist eine Softwaresammlung, die in erster Linie ermöglichen soll unter Linux das DMX4ALL USB-DMX Interface anzusteuern. Inzwischen wird auch das LAN Interface des Herstellers unterstützt und es sollen noch weitere Geräte folgen. Der aktuelle Funktionsumfang beinhaltet einen Steuerungsserver, einige Clientanwendungen für die Kommandozeile, sowie ein programmierbares Steuerungsprogramm.

Durch die Implementation des Treibers als Server werden einfache Fernwartung und vielseitige Betriebsarten möglich. Über Sockets ist es auch sehr einfach die Hardware von anderen Programmiersprachen oder Telnet aus zu steuern. So kann zum Beispiel spielend leicht eine Java-Anwendung, eine PHP-Webseite oder - durch das Tool "dmxset" - auch ein Shellscript die Kontrolle übernehmen.

1.2 Installation: Debian, Ubuntu, Raspbian

Laden Sie den aktuellen .deb-Installer von LINDMX herunter¹. Installieren Sie das Package, indem Sie es unter Ubuntu doppelt anklicken oder in der Kommandozeile über:

```
sudo dpkg --install dateiname.deb
```

„dateiname.deb“ ist dabei durch den Namen der heruntergeladenen Datei zu ersetzen.

Sollte es Probleme mit Libraries geben, versichern Sie sich mit dem folgendem Befehl, dass die benötigten Pakete installiert sind:

```
sudo apt-get install libftdi1 libstdc++6 readline-common libreadline6  
libasound2 libncurses5
```

1.3 Installation: openSUSE

Laden Sie den aktuellen .rpm-Installer von LINDMX herunter². Installieren Sie das Package in der Kommandozeile über:

```
sudo rpm -i dateiname.rpm
```

„dateiname.rpm“ ist dabei durch den Namen der heruntergeladenen Datei zu ersetzen.

Sollte es Probleme mit Libraries geben, versichern Sie sich mit dem folgendem Befehl, dass die benötigten Pakete installiert sind:

```
sudo zypper install libftdi1 libstdc++6 libncurses5 libasound2
```

1.4 Deinstallation

lindmx wird entfernt über:

```
sudo dpkg -r lindmx
```

Bzw. in openSUSE über:

```
sudo rpm -e lindmx
```

¹Downloadseite: <http://www.qikgrp.com/lindmx>

²Downloadseite: <http://www.qikgrp.com/lindmx>

1.5 Inbetriebnahme: DMX4ALL Micro USB

Beim DMX4ALL Micro USB genügt es, das Interface mit dem Computer zu verbinden.

1.6 Inbetriebnahme: DMX4ALL LAN

Zur Inbetriebnahme des DMX4ALL LAN muss zunächst die Entscheidung getroffen werden, ob die Kommunikation über UDP oder TCP stattfinden soll. Bei TCP ist sicher gestellt, dass jedes abgeschickte Paket auch angekommen ist. UDP hingegen prüft nicht, ob das Paket angekommen oder das Interface überhaupt auffindbar ist. Dafür ist UDP aber wesentlich schneller und bei einer größeren Lightshow fällt das ein oder andere Fehlpaket auch sicher nicht auf, während TCP schnell das Netzwerk belastet. Darüber hinaus läuft der LINDMX Server mit der UDP-Variante sicherer, da ausbleibende Antworten des Interfaces ihn nicht blockieren. So macht es auch nichts, wenn das Interface mal zeitweise nicht erreichbar ist.

Die Empfehlung lautet also UDP. Dazu muss zunächst das DMX4ALL LAN Interface richtig konfiguriert werden. Rufen Sie mit einem Browser das Webinterface auf. Dies ist unter der IP-Adresse des Gerätes³. Dort kann nun unter „Channel 1“ und „Connection“ anstatt „TCP“ das Protokoll „UDP“ gewählt werden. „Datagram Type“ sollte auf „00“ gestellt sein. Wichtig: Nachdem Sie auf „OK“ geklickt haben müssen Sie noch mit „Apply Settings“ links im Menü die neuen Einstellungen übernehmen!

Für die Verwendung von TCP ist das Gerät normalerweise bereits richtig eingestellt. Ansonsten verfahren Sie wie oben beschrieben und wählen Sie TCP. Sollte die Webschnittstelle sich über einen fehlenden „Remote Host“ beschweren, tragen sie dort einfach „0.0.0.0“ ein.

Damit der Server das Interface nun auch nutzen kann, müssen Sie noch das Interface in der Konfigurationsdatei einstellen. Am schnellsten geht das in der Konsole mit dem Texteditor nano:

```
sudo nano /etc/lindmxd.conf
```

Der Eintrag „interface“ muss dann auf „dmx4all_udp“ bzw. „dmx4all_tcp“ gesetzt werden. Zusätzlich sind „interface_host“ und „interface_port“ zu setzen. Das kann dann beispielsweise aussehen wie folgt:

```
interface dmx4all_udp
interface_host 192.168.0.250
interface_port 10001
```

Über die Konfigurationsdatei folgt ein eigener Abschnitt.

1.7 Benutzung: Starten des Servers

Zunächst muss der LINDMX Server gestartet werden. Ab Version 0.1.3 erfolgt das am besten über das mitgelieferte Startskript:

```
sudo service lindmxd start
```

bzw. (openSUSE):

```
sudo /etc/init.d/lindmxd start
```

³Voreinstellung: 192.168.0.250, ansonsten gibt es ein Tool von LANTRONIX

lindmxd startet dann mit den Standardeinstellungen (siehe Kapitel „Konfigurationsdatei“). Um den Server wieder zu beeden wird „start“ durch „stop“ ersetzt. Beim Herunterfahren des Computers wird der Dienst automatisch vom System angehalten.

Ob der Server erfolgreich gestartet ist kann geprüft werden mit:

```
sudo service lindmxd status
```

Aufgetretene Fehler werden, wie bei Serveranwendungen üblich, im System Log protokolliert. Um die letzten Fehler anzuzeigen kann unter den meisten Betriebssystemen folgender Befehl verwendet werden:

```
tail /var/log/syslog | grep lindmxd
```

Unter SUSE Linux findet sich die Ausgabe in der Regel unter:

```
tail /var/log/localmessages | grep lindmxd
```

Soll lindmxd zu Testzwecken, zur Fehlersuche oder anderen Zwecken im Vordergrund gestartet werden kann man das Programm auch direkt und mit dem Debug-Flag starten. Beachten Sie, dass Administratorrechte benötigt werden, um auf die serielle Schnittstelle zuzugreifen.

```
sudo lindmxd -d
```

1.8 Benutzung: Steuerung

Läuft der Server, können Kanäle mit dmxset gesetzt werden:

```
dmxset 1 255
```

setzt z.B. den Kanal 1 auf den Wert 255 - also vollständig eingeschaltet. Der Befehl dmxprog (vor 0.1.1 dmxctrl) bietet die Möglichkeit, Chaser (Laufflichter) zu programmieren, gruppieren und abzuspielen. So können kleine „Lightshows“ gebastelt werden. Sowohl dmxset als auch dmxprog können auch von einem anderen Computer benutzt werden (wenn die Firewall es zulässt). Durch den Parameter -s [host] kann ein Rechnername oder die IP des Rechners angegeben werden auf dem lindmxd läuft und das Interface angeschlossen ist. Lautet also die IP-Adresse meines Computers 192.168.178.101, das Interface ist am Computer angeschlossen und dort läuft lindmxd, dann kann das Licht an Kanal 1 z.B. von meinem Laptop aus über mein WLAN gesteuert werden mit:

```
dmxset -s 192.168.178.101 1 255
```

bzw.

```
dmxprog -s 192.168.178.101
```

dmxset bietet darüber hinaus noch die Möglichkeit Daten aus einer Datei zu laden. So kann eine Lichtszene schnell und bequem gesetzt werden. Dazu muss zunächst eine Textdatei angelegt werden, z.B. „szene.dmx“. In diese Datei werden dann Zeilenweise die Kanaldaten geschrieben. Folgender Inhalt von „szene.dmx“ würde Kanal 1 auf 255, Kanal 2 auf 50, Kanal 3 auf 100, ... usw. setzen:

```
2
55
50
100
0
```

```
255
200
0
```

Der dazu passende Aufruf von `dmxset` lautet:

```
dmxset -i scene.dmx
```

1.9 Konfigurationsdatei: `/etc/lindmxd.conf`

Die Konfigurationsdatei des LINDMX-Servers beschreibt das Standardverhalten. So können zum Beispiel die Standardwerte für den Interfacetypen, den Debugmodus etc. gesetzt werden. Dazu wird einfach mit einem Texteditor und root-Rechten die Datei editiert. Hier etwa mit `nano`:

```
sudo nano /etc/lindmxd.conf
```

Natürlich kann statt `nano` auch `gedit` oder `vi` benutzt werden.

Kommentarzeilen beginnen mit einer Raute und werden ignoriert. Die Einstellungsmöglichkeiten sind in der Vorlagendatei ausführlich kommentiert.

1.10 Server beim Systemstart starten

Um den Service beim Systemstart zu starten liegt bereits ein Startskript vor, dieses muss nur noch angemeldet werden. Dazu genügt:

```
sudo update-rc.d lindmxd defaults
```

Möchte man das wieder rückgängig machen, geht das mit:

```
sudo update-rc.d -f lindmxd remove
```

1.11 Sicherheit

LINDMX ist darauf ausgerichtet, in einem geschlossenen und sicheren Netzwerk betrieben zu werden. So soll vor allem die Bedienbarkeit und Portabilität erhöht werden. Daher sollte der LINDMX-Port (Standard: 6767) in der Firewall gegenüber öffentlichen Netzen wie dem Internet stets gesperrt sein. Wenn man es dennoch über das Internet fernsteuern möchte bieten sich andere Möglichkeiten an. Sicher und schnell einzurichten ist ein verschlüsselter SSH Zugang zum Rechner. Wie das geht, ist auf ubuntuusers.de⁴ sehr ausführlich erklärt. Nach dem erfolgreichen Login auf einem Rechner im geschützten System kann man dann wie gewohnt `dmxset` etc. benutzen. Eine weitere Möglichkeit besteht darin, eigene Dienste zu schreiben. So könnte etwa der Apache Webserver mit PHP benutzt werden.

1.12 DMX in Shell-Skripten

Mit `dmxset` können auch Shell-Skripte erwünschte Kanäle setzen. Es folgt ein Beispiel, das eine Lampe im Sekunden-Takt blinken lässt:

⁴<http://wiki.ubuntuusers.de/SSH>

```
#!/bin/sh
i = 10
while [ $i -ge 1 ]
do
    dmxset 1 50
    sleep 1
    dmxset 1 255
    sleep 1
    i = 'expr $i - 1'
done
dmxset 1 0
```

1.13 DMX mit Cronjobs

Mit Cronjobs kann man auf Linux-Systemen automatische Aufgaben ausführen. Das dafür zuständige Programm cron ist auf den meisten Distributionen schon enthalten. Die Aufgaben kann man ganz einfach in der Konsole ändern. Auch das ist bei ubuntusers.de⁵ beschrieben. Durch das Hinzufügen folgender Zeilen wird beispielsweise um 14 Uhr eine Lampe ausgeschaltet und um 18 Uhr wieder eingeschaltet:

```
0 14 * * * dmxset 1 0
0 18 * * * dmxset 1 255
```

⁵<http://wiki.ubuntuusers.de/Cron>

2 dmxprog: Show Programmer

2.1 Einführung

dmxprog ist ein Programm mit dem kleine Lightshows programmiert und abgespielt werden können. Ausgerichtet ist es vor allem auf Effektbeleuchtung zu Musik. Der Start erfolgt in der Konsole über:

```
dmxprog
```

Soll der Server nicht auf dem selben Rechner laufen wie dmxprog, dann kann mit der Option „-s server“ zusätzlich ein Hostname oder eine IP-Adresse übergeben werden. Beispiel:

```
dmxprog -s 192.168.0.100
```

Die Option „-c Zahl“ kann die Anzahl der Kanäle einstellen, auf die zugegriffen wird. Diese Zahl sollte die des Servers nicht überschreiten.

2.1.1 Begriffe

Die folgenden Fachbegriffe aus der Lichtprogrammierung sollten für diese Anleitung bekannt sein.

- **Programmer:**
Funktion zum Erstellen neuer Lichteinstellungen
- **Szene/Step:**
Eine feste Einstellung von Kanalwerten
- **Chaser/Programm:**
Eine Funktion, die im vorgegebenen Rhythmus zwischen Szenen wechselt (z.B. Lauflicht)
- **Fade:**
Überblendezeit zwischen zwei Szenen
- **Kanal:**
Ein Kanal repräsentiert immer eine Funktion (z.B. Helligkeit) eines Geräts
- **Kanalwert/Value:**
Der Wert eines Kanals kann von 0 (dunkel) bis 255 (hell) gehen

2.1.2 Allgemeines

dmxprog gliedert sich in fünf Abschnitte: Programmer, Chasers, Detail, Edit und Global. Im Programmer können neue Lichtszenen geschrieben werden. Unter „Chasers“ wird das Programm, in das geschrieben wird ausgewählt. Das Tab „Detail“ enthält Informationen zum ausgewählten Chaser und kann dessen Einstellungen, wie Geschwindigkeit, Überblendezeit etc. bearbeiten.

„Edit“ dient dazu, die bereits aufgenommenen Szenen des ausgewählten Chaser zu ändern. „Global“ ist ein Bereich, in dem sich übergreifende Funktionen des Programms, wie Speichern und die Kommandozeile befinden.

2.1.3 Konzepte der Wiedergabe

Wichtig bei dmxprog sind vor allem zwei Konzepte. So erhält immer der zuletzt von einem Chaser gesetzte Wert für einen Kanal den Vorrang, nicht etwa der höchste. Darüber hinaus behält ein Kanal seinen Wert so lange - auch über eine Szene hinaus - bis ein Chaser ihm einen neuen Wert zuordnet.

Das zweite wichtige Konzept ist, dass der Programmierer Vorrang vor allem hat. Ist ein Kanal im Programmierer gesetzt, dann kann kein Chaser mehr diesen Kanal überschreiben. Laufende Chaser beeinflussen den Programmierer nicht. Sie werden im Hintergrund abgespielt und nehmen keinen Einfluss auf neue Szenen. Das ermöglicht während einer laufenden Show noch Bestandteile nachträglich zu ergänzen.

2.1.4 Programmieren von Chasern

- Wechseln sie mit der Tabulator-Taste in den Tab „Chasers“
- Wählen Sie mit den Pfeiltasten das Programm aus, in das geschrieben werden soll
- Wechseln Sie zurück in den Tab „Programmierer“
- Wählen Sie mit den Pfeiltasten den Kanal aus, der beeinflusst werden soll
- Setzen sie den Kanalwert mit den Tasten +/-, Bild Auf, Bild Ab oder direkter Zahleneingabe auf den gewünschten Wert
- Wiederholen Sie die zwei vorherigen Schritte, wenn sie weitere Werte beeinflussen wollen
- Speichern sie den Step / die Szene mit Shift+S (Store)
- Wiederholen Sie die letzten Schritte für weitere Steps, bis alle Schritte des Chasers eingegeben sind

Tipps:

- Wichtig: „Ungesetzt“ (durch drei Minuszeichen angezeigt) ist etwas anderes als der Wert Null. Wenn der Wert ungesetzt ist, beeinflusst ein Step den Kanal nicht. Ist er Null, so wird der Kanal beim Aufruf auf Null gesetzt!
- Um einen Kanal wieder auf „Ungesetzt“ zu stellen, kann Strg+U (Unset) gedrückt werden
- Um alle Kanäle zurück zu setzen, kann Strg+C (Clear Programmierer) gedrückt werden
- Häufig muss nach dem Programmieren der Programmierer erst zurück gesetzt werden (Clear), um beim Abspielen des Programmes ein Resultat zu sehen
- Oben unter „Global“ deutet der Text „CLEAR PROG!“ an, dass noch Werte im Programmierer gesetzt sind

2.1.5 Einstellungen von Chasern

- Wechseln sie mit der Tabulator-Taste in den Tab „Chasers“
- Wählen Sie mit den Pfeiltasten das Programm aus, dessen Einstellungen bearbeitet werden sollen
- Wechseln Sie in den Tab „Detail“
- Wählen Sie mit den Pfeiltasten die Einstellung aus (Name, Fade Time, ...), die bearbeitet werden soll
- Verändern Sie den Wert den Tasten +/- (oder bei Texteingaben wie beim Namen mit Enter)

Tipps:

- Benutzen Sie Gruppen, um Geräte und Eigenschaften zu gruppieren. Startet ein Chaser einer Gruppe, werden alle anderen Programme der gleichen Gruppe beendet.
- „Trigger“ ist der Auslöser, der zum Wechseln einer Szene führt. In späteren Versionen soll auch noch eine individuelle Steuerung pro Schritt möglich werden.
- Measure ist die Geschwindigkeit, mit der die Schritte gewechselt werden. Sie ist zusätzlich abhängig vom Beat. Dieser kann mit der Leertaste vorgegeben werden.
- Die Programme 1-26 können zusätzlich auch mit den Tasten A-Z gestartet und beendet werden.

2.1.6 Bearbeiten von aufgenommenen Szenen

- Wechseln sie mit der Tabulator-Taste in den Tab „Chasers“
- Wählen Sie mit den Pfeiltasten das Programm aus, dessen Schritte bearbeitet werden sollen
- Wechseln Sie in den Tab „Edit“
- Bewegen sie sich mit den Tasten „,“ und „.“ durch die Schritte
- Bearbeiten Sie die gewünschten Werte wie im Programmer
- Strg+S muss nicht (!) gedrückt werden!

2.2 Die Kommandozeile

dmxprog verfügt über eine Konsole, die viele Funktionen zusätzlich steuern kann. Auch Stapelverarbeitungsdateien können ausgeführt werden. So lassen sich einige Aufgaben in deutlich weniger Zeit erledigen.

Die Konsole befindet sich im Tab „Global“. Die Eingabe wird mit Enter begonnen und mit „exit“ kann die Konsole verlassen werden.

2.2.1 Allgemeine Befehle

Sende Tap-Signal:

```
tap
```

Sende Sync-Signal:

```
sync
```

Zurücksetzen des Programmers:

```
clear
```

Speichern des Programmers in das aktuell ausgewählte Programm:

```
store
```

Leeren der Konsole:

```
cls
```

Stapelverarbeitungsdatei / Skript ausführen:

```
exec <file >
```

Beats Per Minute setzen:

```
setbpm <number>
```

Ausgabe von Text:

```
echo <text> ...
```

Konsoleneingabe beenden:

```
exit
```

2.2.2 Befehle zur Steuerung der Kanäle

channel kann stets mit c abgekürzt werden. range ist eine Zahl oder ein Bereich wie z.B. 10-16 (für die Kanäle 10 bis 16). Die Schlüsselwörter EVEN sowie ODD sind optional, um nur gerade oder ungerade Werte zu setzen.

```
channel <range> [even/odd] at <value>
```

```
channel <range> [even/odd] on
```

```
channel <range> [even/odd] off
```

```
channel <range> [even/odd] clear
```

```
channel <range> shift
```

```
channel <range> shiftl
```

2.2.3 Befehle zur Steuerung der Chaser

chaser kann stets mit chs abgekürzt werden.

```
chaser <number> on
chaser <number> off
chaser <number> toggle
chaser <number> faster
chaser <number> slower
chaser <number> select
chaser <number> rename <name> ...
chaser <number> group <number>
chaser <number> grp <number>
```

2.2.4 Befehle zur Einrichten von MIDI-Fernsteuerung

Wesentliche Funktionen (Abkürzungen in Klammern):

- Mit dem Standardinterface verbinden: midiconnect (mcon)
- Für ein anderes Gerät ALSA-Kennung hinter midiconnect angeben (z.B. hw:1,0,0)
- midibind (mbind) bindet MIDI-Ereignisse an Befehle
- Wird Status/Datenbyte nicht angegeben, wird der Befehl dem letzten eingegangenen Signal zugeordnet
- Will man also automatisch lernen, dann kann man den Regler betätigen und dann mbind ohne MIDI-Bytes aufrufen
- Im Befehl kann der Velocity-Wert des MIDI-Signals eingesetzt werden indem man ein % einfügt und auf 0-255 skaliert
- Durch eine Zahl direkt hinter % wird der Wert passend skaliert (z.B. %123 für Zahlen von 0 bis 123)
- Bei einem \$ wird ON (bei Wert größer 93) eingesetzt oder OFF (bei Wert kleiner gleich 93)

```
mcon
midiconnect
mcon <device>
midiconnect <device>
mbind : <command> ...
midibind : <command> ...
mbind <statusbyte> <datenbyte1> : <command> ...
midibind <statusbyte> <datenbyte1> : <command> ...
```