

mtrainer

Beispiel für das komplexe Setup von **ALSA/MIDI** Geräten und Programmen.

Inhaltsverzeichnis

Programm mtrainer	1
Hauptmenü mtrainer	1
Unit-Texte: Beschreibung	3
Setup: ALSA MIDI und Unit	4
Setup Beispiel: Virtuelles Keyboard 'vmpk' mit Synthesizer 'timidity'	5
Setup Beispiel: Virtuelles Keyboard 'vmpk' mit Synthesizer 'qsynth'	6
Setup Beispiel: Reales Keyboard mit eingebautem Tongenerator 'psr975'	7
Setup Beispiel: Reales Keyboard 'axiom' ohne Tongenerator mit Synthesizer 'timidity'	8
Test Midi	9

Programm **mtrainer**

Das Programm Musiktrainer kann zum Gehörtraining mit Keyboards oder anderen Midi-Eingabegeräten benutzt werden. Es können verschiedene gespeicherte Tonfolgen über Midi abgespielt werden und die Midi-Eingaben ausgewertet werden. Die selbst erstellten Übung-Units werden als einfache Textdateien gespeichert. Der Übungsablauf kann mit Tasten vom Keyboard aus gesteuert werden. Dafür werden die tiefsten Tasten des Keyboards verwendet.

Play **Key1:** Die letzte Tonfolge der Unit unverändert wiederholen
Pause **Key2:** Übung zum Spielen unterbrechen, die Funktionstasten bleiben aktiv

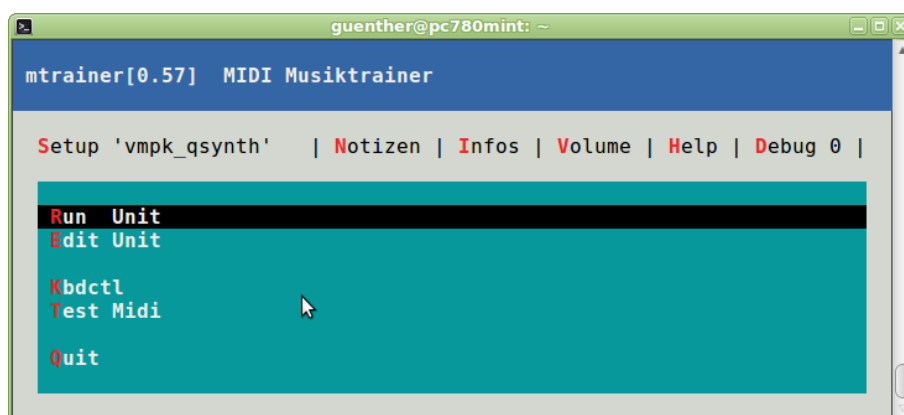
a

Das Programm besitzt ein aufwendiges [Setup](#) für ALSA MIDI-Funktionen, Programme und Geräte.

- Keyboard Verbindungen direkt (raw) oder virtual
- ALSA Connections virtual
- Synthesizer Programm starten
- Virtuelles Keyboard Programm starten
- JACK Connection Kit starten

Alle Einstellungen und ALSA Verbindungen werden beim Programmstart wieder hergestellt. Die eingestellten Hilfsprogramme werden gestartet oder gestoppt.

Hauptmenü **mtrainer**



Die Befehlstasten:

Setup 'vmpk_qsynth' : Setupmenü aufrufen. Das aktuelle Setup ist '!...'.
Notizen : Textdatei für Notizen.
Infos : Programminformationen anzeigen.
Volume : Option: ALSA Volume.
Help : Hilfedateien anzeigen.
Debug : Debugmodus 0/1.

Run Unit : Eine Übung wählen und starten.
Edit Unit : Übungstext bearbeiten.
Kbdctl : Programm Songverwaltung aufrufen.
Test Midi : Midi In- und Output testen.

Unit-Texte: Beschreibung

Die Übungs-Units sind Texte mit C ähnlicher Syntax.

Kommentar: `// xxx ...`
 String-Variablen: `Info="xxxx xxx ...";`
 Integer-Variablen: `Kanal=0;`

Beispiel einer Unit:

```
// Unit 14 =====
// Note c4 und zufällige Note aus c4-g4 spielen
// -----

Info="C-Dur, Note c4 und Zufall c4-g4";

Kanal=0;           // Unit Kanal 0
Anschlag=50;      // Unit Anschlagstärke 0-127
BPM=100;          // Beats pro Minute
n=1;              // Takt: n/m
m=3;

Zufall="c-dur 4"; // Zufallsnoten aus C-Dur4
x      ="Zufall"; // Variable x wird durch Zufallnote ersetzt

Song="c4 - 1/2 x"; // Noten des Songs
                // Note c4 Dauer 1/4, Pause - Dauer 1/4, Note x Dauer 1/2

OkNote  ="x";           // richtige Antwortnote.
OkAktion="Ok";         // richtige Antwort löst Key-Event 'Ok' aus
OkSong  ="1/4 - 1/8 x - x 1/4 -"; // Antwortsong bei Ok
```

Noten, Notenlängen, Akkorde und Pausen:

Mittle-C: c4
 tiefste Note: c0
 höchste Note: b9
 Versetzungszeichen: # oder b
 eingestrichene Oktave: c4 c#4 d4 d#4 e4 f4 f#4 g4 g#4 a4 a#4 b4
 c4 db4 d4 eb4 e4 f4 gb4 g4 ab4 a4 bb4 b4
 Akkord: a(c4|4|7) Grundton mit Halbtonschritten, ergibt c4 e4 g4
 Pausen: -
 Viertelnote c4: 1/4 c4 Die Dauer gilt auch für die folgenden Noten/Pausen
 halbe Pause: 1/2 - Die Dauer gilt auch für die folgenden Noten/Pausen

Festlegung des Notenvorrats, aus dem die zufälligen Noten gewählt werden:

`Zufall=" <Option> , <Option>, usw.";` Die verschiedenen Optionen werden durch Beistriche getrennt.

Beispiele für mögliche Optionen:
`c-dur 5` Grundton c5, C-Dur Tonleiter dazu
`g-dur 3-4` Grundton g3, G-Dur von g3 bis f#5 dazu
`f3 g5 g#5 a4` Einzelne Noten f3 g5 g#5 a4 dazu
`ohne g4 f7` Einzelne Noten aus dem Bereich entfernen.

Beispiele: `Zufall=" c-dur 5, ohne c5";` Noten aus C-Dur von d5 bis b5, zufällig

Variablen für Zufallsnoten definieren: `x = Zufall;` Variablennamen von **i bis z**.

Notenfolgen werden als Songs bezeichnet:

`Song = "c4 - 1/2 x";` Note c4 Dauer 1/4, Pause Dauer 1/4, Zufallsnote aus dem Zufallsbereich Dauer 1/2

Auswertung:

Richtige Antwortnote festlegen: `OkNote = "x";`
 Die richtige Antwort löst Key-Event 'Ok' aus: `OkAktion = "Ok";`
 Antwortsong bei Ok `OkSong = "1/4 - 1/8 x - x 1/4 -";`

Setup: ALSA MIDI und Unit

Die Setup-Befehle:

Das Programm kann verschiedene Setup-Vorlagen verwalten.

Alle Einstellungen und ALSA Verbindungen werden beim Programmstart wieder hergestellt.

Die eingestellten Hilfsprogramme werden gestartet oder gestoppt.

```

guenther@pc780mint: ~
Setup | Datei: mtrainer.conf

Setup: vmpk_qsynth | Setupvorlagen speichern oder laden
Keyboard 'vmpk' mit Synthesizer 'qsynth' | Beschreibung
mtrainer.conf | Dateiname

Programs: Synthesizer qsynth &
Jack qjackctl &
Keyboard vmpk &
Volume alsamixer

Input : ALSA 'virtual' | NONBLOCK [ok]
'VMPK Output' '>'Virtual RawMIDI '
Filter: NoClock NoSense

Output : ALSA 'virtual' | APPEND [ok]
'Virtual RawMIDI' '>'Synth input port
Delays ns: Byte=35211 | Event=352112

Keyboard: ALSA 'VMPK Output' '>'Synth input port

ALSA Infos | set Connections | List Devices/Clients

Unit: Infos anzeigen
Midi: Key1='c1' | Key2='d1' | Kanal=0 | Anschlag=64

RETURN set | Befehl | ESC ? █

```

Setup : Setupvorlagen speichern oder laden.
Programs : Hilfsprogramme automatisch starten und beenden.
 Programme mit '-' vor dem Namen werden nicht gestartet.

Input : ALSA Input raw oder virtual. Für virtual ALSA Connections. Midifilter.
Output : ALSA Output raw oder virtual. Für virtual ALSA Connections. Mididelays.
Keyboard : Option ALSA Connection: Für virtuelles Keyboard (vmpk) oder Synthesizer
ALSA Infos : ALSA Einstellungen anzeigen.
Set Connections : ALSA Connections manuell herstellen.
List Devices/Clients : ALSA Devices, Clients und Connections anzeigen.

```

guenther@pc780mint: ~
List ALSA Raw-Devices | ALSA Clients | ALSA Connections

RawMIDI Devices - Input
virtual

RawMIDI Devices - Output
virtual

ALSA Clients - Input
14:0 'Midi Through Port-0'
129:0 'VMPK Input'
131:0 'Virtual RawMIDI'
133:0 'Synth input port'

ALSA Clients - Output
0:0 'Timer'
0:1 'Announce'
14:0 'Midi Through Port-0'
128:0 'VMPK Output'
132:0 'Virtual RawMIDI'

ALSA Connections
128:0 'VMPK Output' verbunden zu: 131:0, 133:0
132:0 'Virtual RawMIDI' verbunden zu: 133:0

Ende mit q

/tmp/less4537 lines 1-28/28 (END)

```

Unit: Infos anzeigen : Infos für die geladene Unit anzeigen.
Midi : Key1: Übung wiederholen | Key2: Pause.
RETURN set : Einstellungen anwenden und Programme starten/stoppen

Setup Beispiel: Virtuelles Keyboard 'vmpk' mit Synthesizer 'timidity'

Virtuelles Keyboard mit Software-Synthesizer.

Die Hilfsprogramme:

Virtuelles Keyboard vmpk:

`sudo apt-get install vmpk`



Software-Synthesizer timidity:

`sudo apt-get install timidity`

Vorlage: `vmpk_qlsynth`

Input: `ALSA 'virtual'`

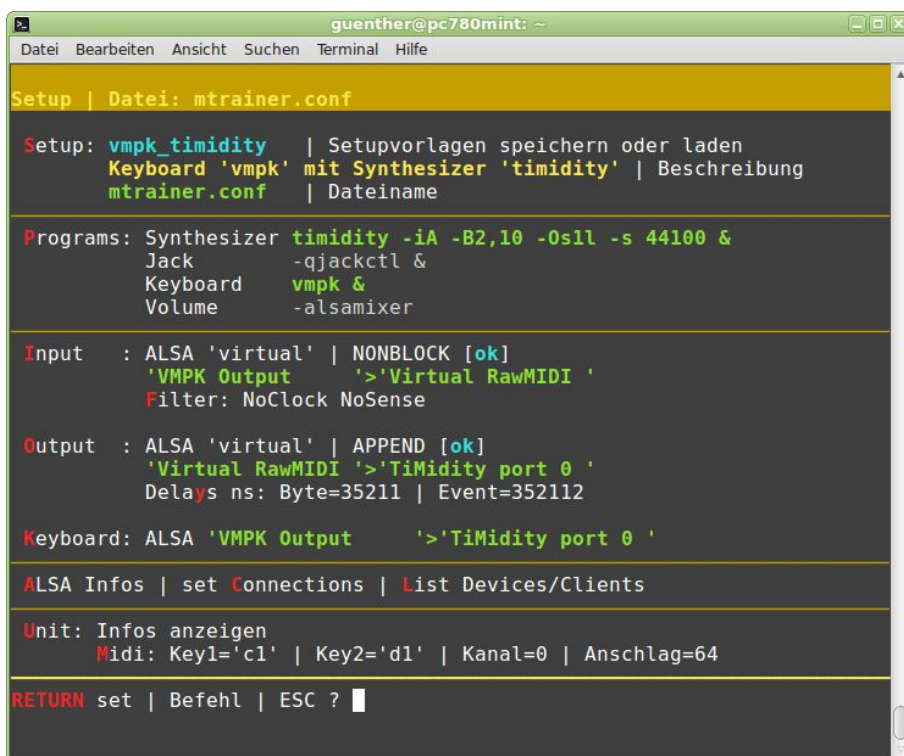
Output: `ALSA 'virtual'`

Die ALSA Verbindungen:

Keyboard vmpk mit Programm
`'VMPK Output'>'Virtual RawMIDI'`

Programm mit Synthesizer
`'VMPK Output'>'TiMidity port 0'`

Keyboard vmpk mit Synthesizer
`'VMPK Output'>'TiMidity port 0'`



Setup Beispiel: Virtuelles Keyboard 'vmpk' mit Synthesizer 'qsynth'

Virtuelles Keyboard mit Software-Synthesizer.

Die Hilfsprogramme:

Virtuelles Keyboard **vmpk**:

Software-Synthesizer **qsynth**:
`sudo apt-get install qsynth`

Option: **jackctl**:
 Zur Anzeige der ALSA Verbindungen.

Die Verbindungen:

Vorlage: **vmpk_qsynth**

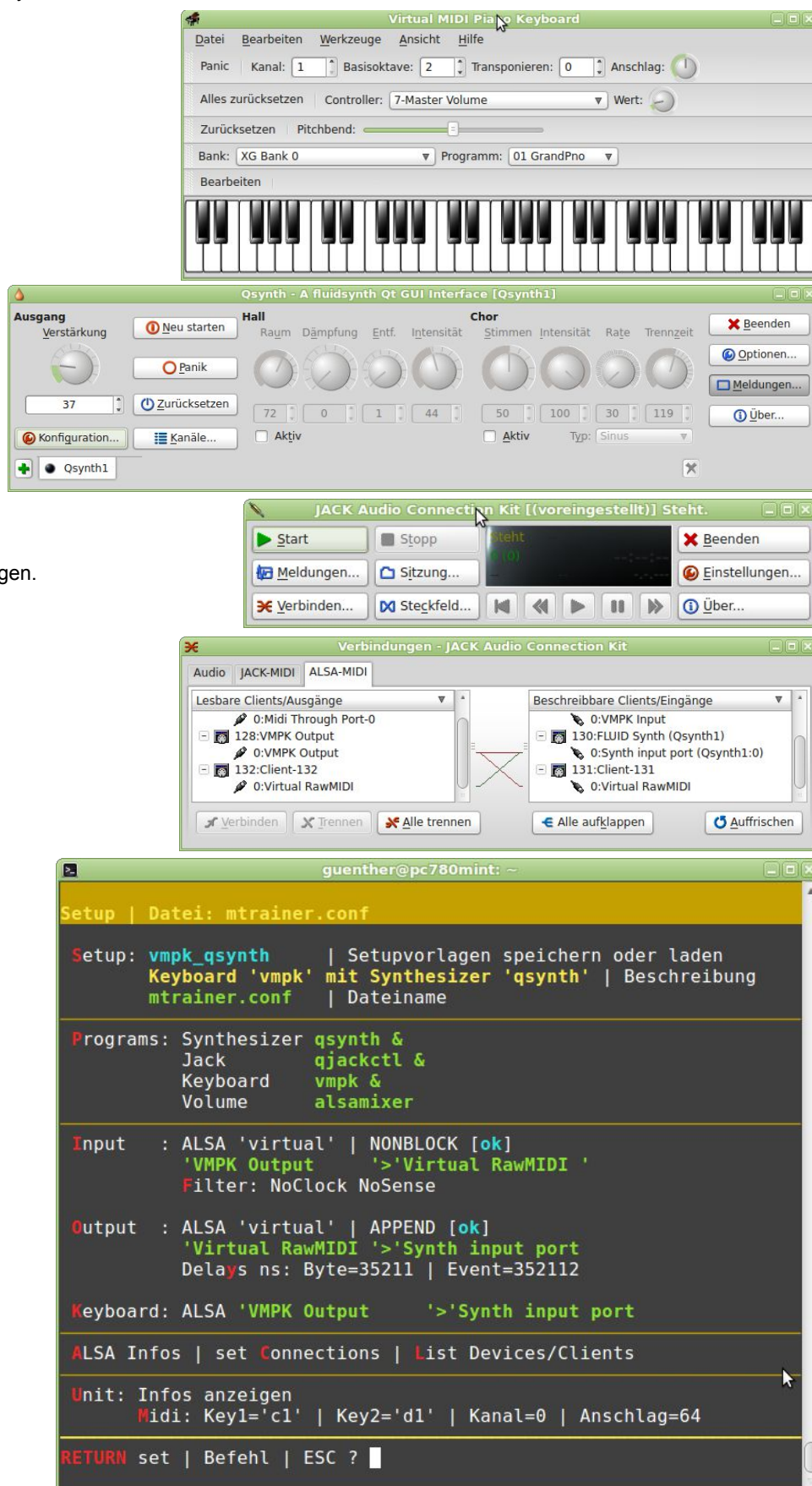
Input: **ALSA 'virtual'**
 Output: **ALSA 'virtual'**

Die ALSA Verbindungen:

Keyboard vmpk mit Programm
 'VMPK Output' '>' 'Virtual RawMIDI'

Programm mit Synthesizer
 'VMPK Output' '>' 'Synth input port'

Keyboard vmpk mit Synthesizer
 'VMPK Output' '>' 'Synth input port'



Setup Beispiel: Reales Keyboard mit eingebautem Tongenerator 'psr975'

Reales Keyboard mit Ton ohne Software-Synthesizer.

Keyboard psr975:



Raspberry Pi 3 B+:



Einstellungen:

Vorlage: psr975

Keine Programme.

Input: ALSA-Raw

Output: ALSA Raw

Kein Keyboard.

```

guenther@pc780mint: ~
Setup | Datei: mtrainer.conf

Setup: psr975 | Setupvorlagen speichern oder laden
Keyboard 'psr975' ohne Synthesizer | Beschreibung
mtrainer.conf | Dateiname

Programs: Synthesizer -timidity -iA -B2,10 -Os1l -s 44100 &
Jack -qjackctl &
Keyboard -vmpk &
Volume -alsamixer

Input : ALSA 'Digital Keyboard MIDI 1' | NONBLOCK [ok]
Filter: NoClock NoSense

Output : ALSA 'Digital Keyboard MIDI 1' | 0 [ok]
Delays ns: Byte=35211 | Event=352112

Keyboard: ALSA -'VMPK Output '>'TiMidity port 0 '

ALSA Infos | set Connections | List Devices/Clients

Unit: Infos anzeigen
Midi: Key1='c2' | Key2='b2' | Kanal=0 | Anschlag=100

RETURN set | Befehl | ESC ? █
    
```

Setup Beispiel: Reales Keyboard 'axiom' ohne Tongenerator mit Synthesizer 'timidity'

Reales Keyboard ohne Ton ohne Software-Synthesizer.



Raspberry Pi 4 B:

Keyboard: Axion:

Hilfsprogramm:

Software-Synthesizer **timidity**:
`sudo apt-get install timidity`

Vorlage: [vmpk_qsynth](#)

Input: **ALSA 'virtual'**
 Output: **ALSA 'virtual'**

Die ALSA Verbindungen:

Keyboard mit Programm
`'Axiom 49 MIDI 1 '>'Virtual RawMIDI '`

Programm mit Synthesizer
`'Virtual RawMIDI '>'TiMidity port 0 '`

Keyboard mit Synthesizer
`'Axiom 49 MIDI 1 '>'TiMidity port 0 '`

```

guenther@pc780mint: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe

Setup | Datei: mtrainer.conf

Setup: axiom_timidity | Setupvorlagen speichern oder laden
Keyboard 'AXIOM' mit Synthesizer 'timidity' | Beschreibung
mtrainer.conf | Dateiname

Programs: Synthesizer timidity -iA -B2,10 -Os1l -s 44100 &
          Jack -qjackctl &
          Keyboard -vmpk &
          Volume alsamixer

Input : ALSA 'virtual' | NONBLOCK [ok]
       'Axiom 49 MIDI 1 '>'Virtual RawMIDI '
       Filter: NoClock NoSense

Output : ALSA 'virtual' | APPEND [ok]
        'Virtual RawMIDI '>'TiMidity port 0 '
        Delays ns: Byte=35211 | Event=352112

Keyboard: ALSA 'Axiom 49 MIDI 1 '>'TiMidity port 0 '

ALSA Infos | set Connections | List Devices/Clients

Unit: Infos anzeigen
Midi: Key1='c2' | Key2='d2' | Kanal=0 | Anschlag=64

RETURN set | Befehl | ESC ?
    
```

Test Midi

Die MIDI-Verbindungen tesen:

```

guenther@pc780mint: ~
ALSA-MIDI: Verbindungen testen

Read      MIDI Input
Write     MIDI Output

Messages  MIDI Definitionen anzeigen

Befehl | ESC ? █

```

Read MIDI Input:

```

guenther@pc780mint: ~
Read Events | Print Bytes mit Info

Save Dump | Ende | Clear Screen | Quit
Dump Name: '/home/guenther/c/bin/mtrainer/bi

90 37 41 |NoteOn|Key|Vel
80 37 00 |NoteOff|Key|Vel
90 34 41 |NoteOn|Key|Vel
80 34 00 |NoteOff|Key|Vel
90 34 41 |NoteOn|Key|Vel
80 34 00 |NoteOff|Key|Vel
█

```

Write MIDI Output:

```

guenther@pc780mint: ~
MidiSong: 0x8903200

Songname      : Tonleiter
MIDI Kanal    : 0
MIDI Anschlag : 64
msNote 1/1    : 2400
msNoteNxt     : 600
EvLst         : 14
  0 | 90 3C 40
 600 | 90 3C 00
 600 | 90 3E 40
1200 | 90 3E 00
1200 | 90 40 40
1800 | 90 40 00
1800 | 90 41 40
2400 | 90 41 00
2400 | 90 43 40
3000 | 90 43 00
3000 | 90 45 40
3600 | 90 45 00
3600 | 90 47 40
4200 | 90 47 00
I    : -1

Start 'Tonleiter' ms=0
MidiSongPlayBis: 'Tonleiter' msBis=0 Size=14 I=0
RawMidiWriteEv: 0 | 90 3C 40
MidiSongPlayBis: 'Tonleiter' msBis=600 Size=14 I=1
RawMidiWriteEv: 600 | 90 3C 00

```


Das Linux-Programm wurde mit Projekt c/ entwickelt.

Installation/Anleitung siehe Ordner c/1_read.me.

Lizenz: Open Source, Copyright GNU General Public License:

mtrainer: MIDI Musiktrainer

Copyright (C) 2020

**Günther Schardinger,
Forstweg 12, 8793-Trofaiach, Austria
v.schardinger@gmx.net**

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 3 der Lizenz oder jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, daß es Ihnen von Nutzen sein wird, aber OHNE IRGEND EINE GARANTIE, sogar ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK. Details finden Sie in der GNU General Public License.

Sie sollten ein Exemplar der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, siehe <http://www.gnu.org/licenses/>.
