

Anleitung: Fluglärmmessung mit dem Raspberry Pi

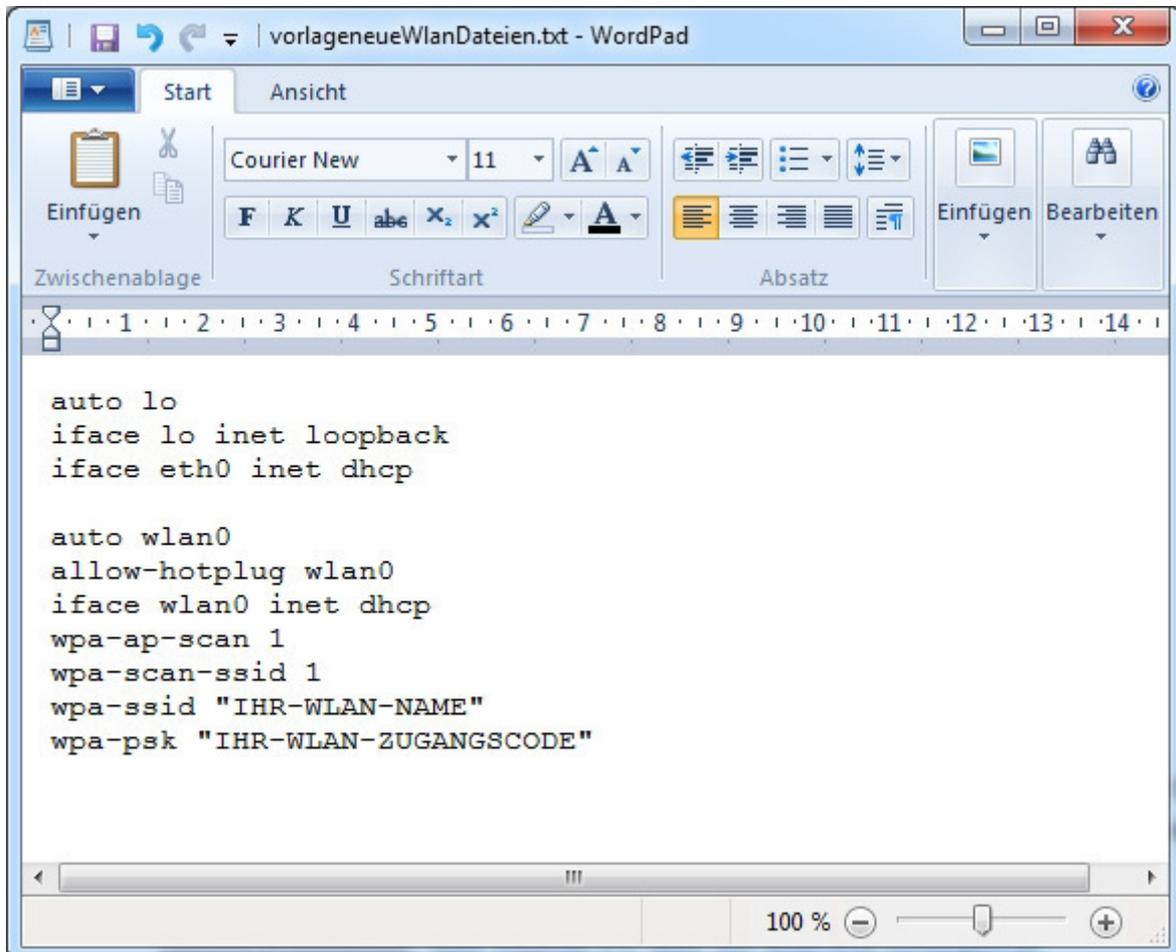
Einrichten einer Messstation

Bevor Sie die Messstation als autonome Einheit in Betrieb nehmen können, müssen Sie Ihre WLAN-Daten und Ihre Zelle.ini Datei an den Raspberry Pi übermitteln. Eine Möglichkeit wäre, Maus Tastatur und Bildschirm anzuschließen und die Konfiguration am Raspberry vorzunehmen. In diesem Text stelle ich allerdings die einfachere Möglichkeit vor die Einstellungen über einen PC mit Hilfe eines SD-Karten-Lesegeräts vorzunehmen.

Dazu nehmen Sie die mitgelieferte MicroSD-Karte und stecken diese in den MicroSD-to-SD-Karten-Adapter. Nun können Sie die Karte in ein SD-Kartenlesegerät einstecken. Greifen Sie über den Datei-Manger auf den Wechseldatenträger zu. Die Bezeichnung für den Datenträger lautet **boot**. Auf dem Datenträger befinden sich zwei Unterordner: **updates** und **backup**. Öffnen Sie den Ordner **updates**.

Einrichtung des WLANs:

In dem **updates** Order finden Sie eine Datei mit dem Namen: **vorlageneueWlanDaten.txt**. Öffnen Sie diese Datei (am besten mit **Wordpad**) und fügen Sie hinter **wpa-ssid** den Namen Ihres Netzwerkes ein und hinter **wpa-psk** Ihren WLAN-Schlüssel (beides in Anführungszeichen). Speichern Sie nun die Datei unter den Namen **neueWlanDaten.txt** in den **updates** Order ab.



The screenshot shows a Windows WordPad window titled "vorlageneueWlanDateien.txt - WordPad". The window contains the following text:

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp

auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-ap-scan 1
wpa-scan-ssid 1
wpa-ssid "IHR-WLAN-NAME"
wpa-psk "IHR-WLAN-ZUGANGSCODE"
```

Bild: Einstellung der WLAN-Daten mit Wordpad

Einrichtung der Zelle.ini:

Sie sollten vom DFLD eine ini-Datei zugeschickt bekommen haben, die Ihre Einstellungen und Zugangsdaten enthält.

Sollte dies nicht der Fall sein, gehen Sie auf die Seite:

<http://dfld.de/Mess/Language/MessStationG2.html>

Dort finden Sie unter Punkt **4. Software** eine E-Mail-Adresse. Schicken Sie dorthin eine Mail mit der Information, dass Sie eine ini-Datei benötigen. In die Mail sollten Sie ebenfalls Ihren Ort und Ihren Straßennamen (ohne Hausnummer) eintragen, damit die Daten richtig eingetragen werden können. Sie bekommen die fertig konfigurierte ini-Datei dann zugesandt.

Kopieren Sie die ini-Datei in den **updates**-Ordner. Nun können Sie den Raspberry Pi zum ersten Mal starten.

Einrichtung der Hardware

Sie können nun die mitgelieferte Hardware an Ihren Raspberry anschließen. Beginnen Sie mit der Messstation, die über den FTDI-Adapter an die USB-Buchse des Raspberry angeschlossen wird. Dann stecken Sie den WLAN-Stick ein. Die MicroSD-Karte kommt in den kleinen Schlitz auf der Seite, die den USB-Anschlüssen gegenüber liegt. Nun können Sie das Micro-USB-Netzteil mit dem Raspberry verbinden und damit dem Raspberry starten.

Es wird nun einige Minuten dauern, bis alle Daten eingestellt wurden. Meist erkennen Sie das vollständige Booten daran, dass die grüne LED auf dem Board nur noch ca. einmal pro Sekunde blinkt. Nun können Sie wie unten beschrieben testen, ob alles funktioniert.

Funktionsstest des Raspberry Pi's

Wenn Sie den Raspberry ohne Bildschirm betreiben ist es oft schwierig zu sagen, ob alles funktioniert hat. Es gibt allerdings die Möglichkeit, die Funktionalität des Raspberry zu überprüfen. Dazu müssen Sie entweder die IP-Adresse Ihres Pi's wissen (indem Sie z.B. in Ihrem Router nachsehen) oder Sie installieren auf Ihrem PC den Bonjour-Service von Apple:

<http://support.apple.com/kb/DL999>

Dieser Service ist dazu da, Geräte die im Netzwerk sind unter einem leichteren Namen auffindbar zu machen. Nach der Installation des Service (Hinweise auf der Seite beachten) können Sie in Ihrem Browser die Seite

<http://raspberrypi.local/>

aufrufen. Falls Sie die IP-Adresse des Raspberry Pi's wissen, können Sie auch diese auch ohne die Installation des Bonjour-Service in dem Browser eintippen.

Auf der Seite finden Sie immer die neueste Log-Datei. Diese Datei wird beim Start des Lärmmessservices angelegt und enthält unter anderem die ersten gemessenen Werte. Somit können

Sie sich sicher sein, dass eine Internetverbindung besteht und dass die Ermittlung der dB-Werte funktioniert.

Tipp: Manchmal kann die Webseite nicht direkt angezeigt werden. Laden Sie die Seite mehrmals neu um zu überprüfen, ob der Raspberry wirklich nicht online ist.

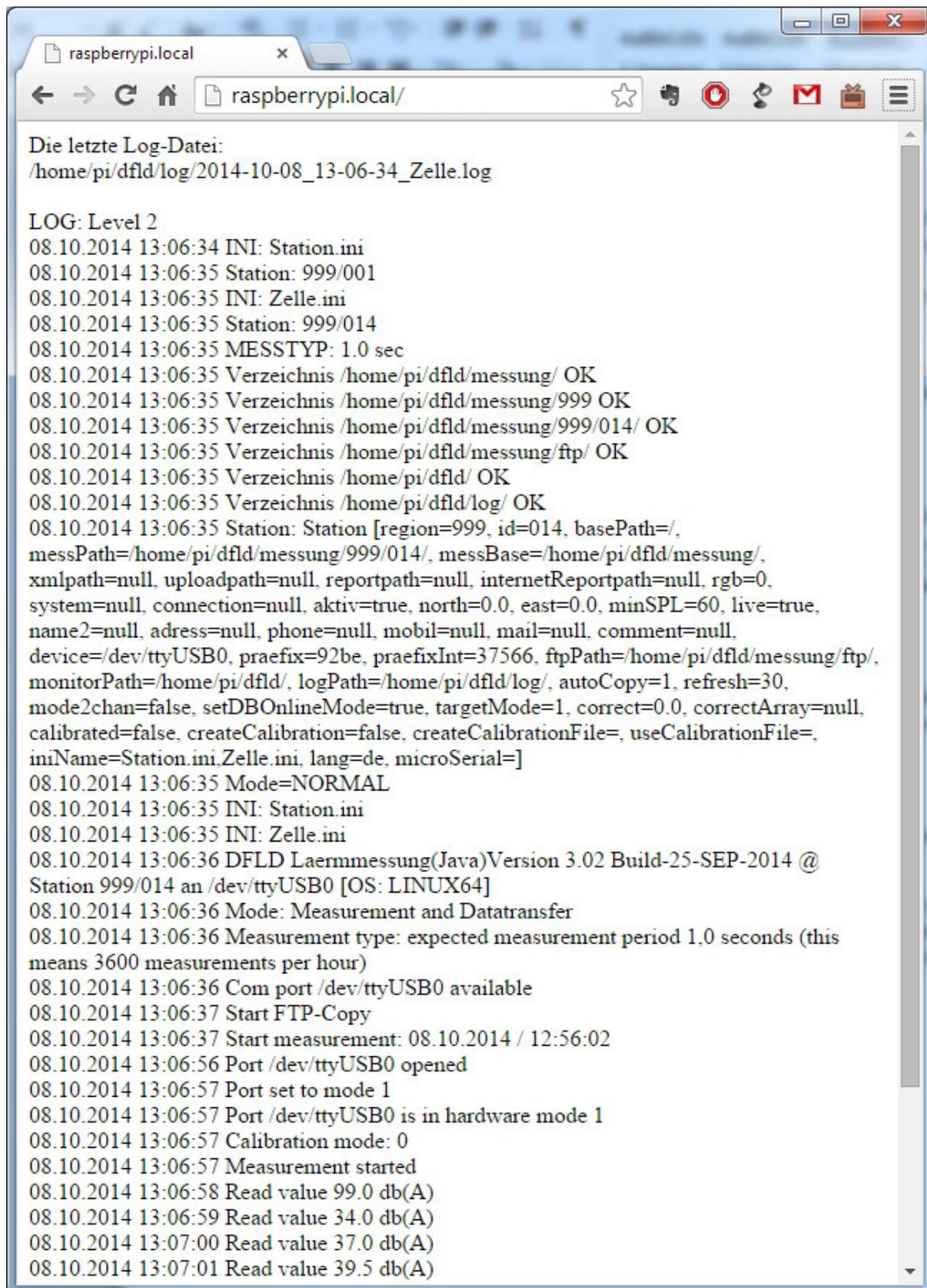


Bild: Die Log-Datei im Browser

Kurzinfos zur Konfiguration:

Betriebssystem: Raspbian (Debian Wheezy) Version September 2014

Hostname: raspberrypi

Benutzername: pi

Passwort: raspberry

SSH-Server: Ja

Installierte Zusatzsoftware:

- Avahi (für Zeroconf)
- DFLD-Software für Debian. Version 3.02 (
- Apache2 Server



29. September 2014

Benutzerhandbuch Lärmmessung mit Zelle 3.0 unter LINUX

Verfasser: Eric Müller
Rückmeldungen/Fragen/Korrekturen/Gute Ideen bitte per Mail an:
fluglaerm_gigu@yahoo.de

Inhaltsverzeichnis

1. Grundlagen des Projektes.....	3
2. Einrichten der Zelle.....	3
Falls eine ältere Software (Zelle 1.0 bis 2.0b) bereits installiert ist.....	3
Automatische Installation (empfohlen).....	4
Manuelle Installation.....	6
Durchführung der manuellen Installation (Expertenmodus).....	7
3. Zelle als Dienst.....	9
4. Erster Start.....	11
5. Auflistung der Schnittstellen.....	12
6. Release Notes.....	14

1. Grundlagen des Projektes

Zelle 3.0 ist die Kontrollsoftware, die ein db-Online Modul an den DFLD Messdienst anbindet. Die Messung erfolgt kontinuierlich. Im Normalfall werden summarische Dateien stündlich sowie Live-Werte 1 Mal pro Sekunde zum DFLD Server übertragen.

Zelle 3.0 läuft im Hintergrund und benötigt sehr wenige Ressourcen, so dass der weiteren Verwendung des Servers als Fileserver, Proxy-Server oder ähnliches nichts entgegensteht.

Die Visualisierung der Daten erfolgt in der vorliegenden Version ausschließlich über die Webseiten des DFLD unter www.dfld.de.

Informationen speziell zur Migration von älteren Version sind in Kapitel 2c zu finden.

2. Einrichten der Zelle

Falls eine ältere Software (Zelle 1.0 bis 2.0b) bereits installiert ist

Da sich die Struktur der Scripts, Verzeichnisse, die Parametrisierung der Programme und die Struktur des Puffer-Files deutlich von der alten Version Zelle 1.1 und 2.x unterscheidet, muss die Installation der neuen Software in einen neuen User oder ein neues Verzeichnis erfolgen.

Hierbei kommt es prinzipbedingt zum Datenverlust der bisher gemessenen Werte des laufenden Tages.

Idealerweise sollte daher die Umschaltung kurz nach Mitternacht erfolgen, wenn der Ausfall möglichst kurz sein soll.

Die Installation der neuen Version kann vorher erfolgen.

Um Mitternacht wird dann die alte Version gestoppt und die neue Version gestartet.

Bis V2.2 nutzte die DFLD Software /etc/init.d/zelle für die Steuerung. Die Beue Version nutzt nun /etc/init.d/dfld

So kann der Wechsel von alte auf neue Installation durch folgende 2 Kommandos bewirkt werden:

1. `sudo /etc/init.d/zelle stop`
2. `sudo /etc/init.d/dfld start`

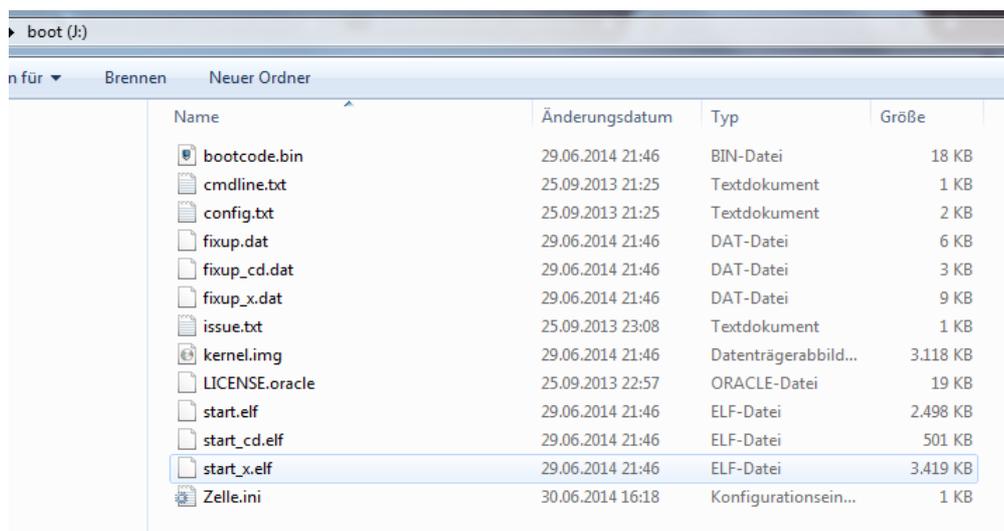
Automatische Installation (empfohlen)

Raspberry-Pi Erweiterung:

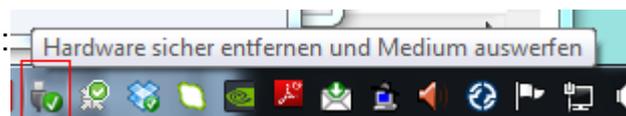
Das /boot Verzeichnis einer aktuellen Raspberry-Wheezy Installation kann als SD-Karte auch von einem PC gelesen werden.

Das nutzen wir um die Installation der vom Admin versendeten Zelle.ini zu vereinfachen, wenn Sie über einen PC mit einem SD-Kartenleser verfügen:

1. Fahren Sie Ihren Raspberry-Pi herunter. (sudo init 0).
2. Wenn nur noch die rote LED leuchtet, trennen Sie das Gerät vom Strom
3. Entnehmen Sie die SD Karte und stecken Sie sie in das SD-Karten-Lesegerät am PC
4. Es wird ein temporäres Laufwerk gemountet, Z.B K:, J:, etc. Der Name das Laufwerks lautet „boot“. Dieses Laufwerk enthält die hier gezeigten Dateien.



5. Kopieren Sie die vom DFLD-Admin erhaltene Zelle.ini in dieses Verzeichnis und „werfen“ Sie das Medium aus :



6. Starten Sie nun wieder den RasPi mit der SD Karte

Durchführung der automatischen Installation

Die automatische Installation ist möglich auf Debian Systemen und Derivaten (z.B. Ubuntu).

Hierbei wird zuerst ein Installationsscript mittels **wget** vom DFLD Server **testapi.dfld.de** geladen, das wiederum die eigentliche Installation nachlädt, installiert und parametrisiert.

Sie haben die Möglichkeit, die vom DFLD-Admin gesendete System.ini Datei während der Installation automatisch in das richtige Verzeichnis kopieren zu lassen.

Befehle (Beispiel eines Raspberry Pi ins Verzeichnis ~/dfld) :

1. pi@raspberrypi ~ \$ mkdir dfld
2. pi@raspberrypi ~ \$ cd dfld
3. pi@raspberrypi ~/dfld \$ wget testapi.dfld.de/downloads/dfld-debian-install.sh
4. pi@raspberrypi ~/dfld \$ bash dfld-debian-install.sh

Das Installationsscript installiert die erforderlichen Bibliotheken, lädt die DFLD-Installation und -Programme nach, parametrisiert die Steuerdateien und legt die System-Startscripts an.

Haben Sie wie im vorherigen Kapitel beschrieben die Datei namens Zelle.ini auf der SD-Karte gespeichert, wird diese ins aktuelle Verzeichnis kopiert.

Falls nicht, können Sie die Datei Zelle.ini jetzt mittels eines File-Transfer-Programms wie WinSCP (<http://www.heise.de/download/winscp.html>) auf den Rechner in das Unterverzeichnis dfld übertragen.

Nun kann die Messung mit

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

gestartet werden.

Manuelle Installation

Voraussetzungen

Das System wurde unter verschiedenen Linux Derivaten getestet. Unter manchen Distributionen sind gegebenenfalls Anpassungen bei dem RC Script notwendig. Folgende Pakete werden genutzt und müssen auf dem System installiert sein.

- NTP-Dienst zur kontinuierlichen Zeitsynchronisation
- java 7. oder höher (Oracle Java oder OpenJDK)
- RXTX Library zur Anbindung der seriellen Schnittstelle

Vorbereitungen

- Für die Zeitsynchronisation sollte das Paket „ntp“ installiert sein.
- Da Zelle 3.0 eine in Java geschrieben ist, wird das Paket Java von SUN benötigt (OpenJDK funktioniert aber auch).
- Zuletzt wird noch die RXTX Library benötigt, mit der Java Programme auf die serielle Schnittstelle zugreifen können.

Unter Ubuntu werden die genannten Programme mit :

```
sudo apt-get install ntp oracle-java7-jdk librxtx-java
```

installiert.

Hinweis: Bei anderen Distributionen, die nicht auf Debian basieren, müssen die genannten Pakete auf anderem Wege installiert werden (z.B. über „yast“ bei SuSE Installationen oder „yum“ bei Red Hat). Die genannten Paketnamen können hierbei leicht variieren.

Bei der Installation der RXTX Library ist darauf zu achten, dass sie zur gewählten Java Runtime und zur HW Architektur passt.

Wenn in Zukunft die Java Installation geändert wird, muß in der Regel auch das Paket RXTX dazu passend wieder eingespielt werden.

Durchführung der manuellen Installation (Expertenmodus)

Wir gehen in diesem Beispiel davon aus, dass die Software unter dem bestehenden User USER im Verzeichnis /home/USER installiert werden soll (Der User selbst ist nicht wichtig und benötigt keine root-Rechte, allerdings erfordern manche Distributionen die Mitgliedschaft des Users in einer bestimmten Gruppe, um auf die serielle Schnittstelle zugreifen zu dürfen).

```
cd /home/USER
mkdir dfld
cd dfld
mkdir tmp
cd tmp
wget testapi.dfld.de/LaermLinux30.zip
cd ..
jar xf tmp/LaermLinux30.zip
```

Nach dem so erfolgten Auspacken des Paketes LaermLinux30.zip ergibt sich folgende Struktur:

```
/home/USER/dfld
  /doc
      dokumentation_linux_zelle.pdf
  /lib
      commons-net-2.2.jar
      dfld-messung.jar
  /log
  /messung
      ftp
  /tmp
      LaermLinux30.zip
  /tools
      dfld.rc.template
      Station.ini.template
```

In das Verzeichnis /home/dfld/zelle muss auch die vom DFLD zugewiesene **Zelle.ini** auf den Mess-Rechner in das Unterverzeichnis dfld kopiert werden, z.B. mittels eines File-Transfer-Programms wie WinSCP (<http://www.heise.de/download/winscp.html>)

Bitte beachten Sie hierbei, dass je nach Art des Transfers am Zeilenende DOS Steuerzeichen auftauchen können. Um das zu überprüfen, öffnen Sie die Datei Zelle.ini mit vi: vi Zelle.ini.

Ist am Ende jeder Zeile ein ^M sichtbar, muss die Datei repariert werden.

Am Einfachsten führen Sie das durch, indem Sie die Datei im ASCII-Modus nochmals transferieren.

Sollte das nicht möglich sein, gehen Sie wie folgt vor (jetzt wird es leider etwas komplizierter)

Geben Sie hierzu folgende Sequenz in den vi ein:

```
:1,$ s/^V^M//g
```

Hierbei befindet sich zwischen dem \$ und s ein Leerzeichen. Weiterhin steht das Zeichen ^ für die Taste CTRL oder STRG. ^V^M ist also ein CTRL-V gefolgt von einem CTRL-M

Die ^M am Ende jeder Zeile sind jetzt verschwunden. Speichern Sie die Datei mit ESC:wq

Der Inhalt von Zelle.ini darf nicht geändert werden, Anpassungen der Umgebung erfolgen ausschließlich in Station.ini.

Kopieren Sie die Stations.ini Vorlage in das aktuelle Verzeichnis:
cp tools/Station.ini.template Station.ini

Die im folgenden rot markierten Einstellungen müssen in Station.ini i.d.R. geändert werden

Konfiguration	Erklärung
[Station]	Stations-Sektion
Device=/dev/ttyUSB0	Das serielle Device an dem die Ak_Modulbus Messtation angeschlossen ist, typischerweise /dev/ttyS0 oder /dev/ttyS1 oder /dev/ttyUSB0
BasePath=/home/dflld/zelle	Pfad in dem die Messdateien abgelegt werden sollen
MssZeit=1	MssZeit=1: Hardware arbeitet im 1sek Takt, (neue Version, DIN Standard) MssZeit=06 Hardware arbeitet im 0,6sek Takt (alte Version)
AutoCopy=1	Sollen Dateien automatisch zum DFLD Server übertragen werden? 1=ja, 0=nein
Refresh=30	Sekunden zwischen der Speicherung der Statusdateien
Language=de	Spracheinstellung: de oder en möglich. Beeinflusst Log-Ausgaben und Fehlermeldungen
[Internet]	Internet Sektion
ExecuteDial=0	Sollen externe Kommandos vor der FTP-Übertragung ausgeführt werden? 0=Nein 1=Ja
DialCommand=/path/to/dial-script.sh	Kommando, das vor der Übertragung ausgeführt werden soll. Hier kann z.B. ein dial-in script aufgerufen werden.
HangupCommand=/path/to/hangupscript.sh	Kommando, das nach der Übertragung ausgeführt werden soll. Hier kann z.B. ein hangup script aufgerufen werden.

Als Device muss der Name der seriellen Schnittstelle stehen, an die db-Online angeschlossen ist. Typischerweise ist das /dev/ttyS0, /dev/ttyS1 oder eins der /dev/ttyUSB* Devices bei Seriell-USB-Adaptoren wie z.B. hier rechts dargestellt (groß-klein Unterscheidung ist bei der Angabe der Devices wichtig, auch hier existieren unterschiedliche Schreibweisen in den einzelnen Distributionen).



Bei der Auswahl der Pfade für Mess, FTP und Log Dateien achten Sie bitte darauf, dass der ausführende User Schreibrechte auf diese Verzeichnisse haben muss.

Falls unklar ist, an welcher Schnittstelle das Mikrofon angeschlossen wurde, finden Sie in Kapitel 5 eine Hilfestellung.

3. Zelle als Dienst

Der Betrieb der Zelle erfolgt als Dienst im Hintergrund. Dazu dient das Script „dflld.rc.template“, welches Sie unter /home/dflld/zelle/tools finden und das sich in die Startscripte integrieren lässt. Es weist die üblichen Parameter stop, start, restart und status auf.

Passen Sie in dieser Datei die rot markierten Parameter an. Kopieren Sie danach als root (bzw mittels sudo) die Datei nach /etc/init.d/dflld mittels:

```
„sudo cp tools/dfld.rc.template /etc/init.d/dfld“
```

javapath=/usr/

Den Java Pfad „javapath“ finden Sie mit „which java“. Von der Ausgabe müssen Sie „/bin/java“ abziehen. Sollte hier also /usr/bin/java ausgegeben werden, müssen Sie /usr/ eintragen. /usr/ ist typischerweise richtig.

homepath=__BASE_PATH__

„homepath“ ist das Verzeichnis, in das Sie installiert haben. In unserem Beispiel also homepath=/home/dfld/zelle/

classpath=lib/dfld-messung.jar:lib/commons-net-2.2.jar:__RXTX_JAR__

Die Datei RXTXcomm.jar finden Sie mit „find / -name RXTXcomm.jar 2>/dev/null“
Lautet die Ausgabe z.B. /usr/share/java/RXTXcomm.jar ändern Sie die Zeile oben folgendermaßen:

```
classpath=lib/dfld-messung.jar:lib/commons-net-2.2.jar:/usr/share/java/RXTXcomm.jar
```

__JNI_PATH__ Hier muss der Pfad zu den JNI Bibliotheken bestimmt werden.

Ermitteln Sie hierzu die Lokation der librxtxSerial.so:

```
pi@raspberrypi ~/dfld $ find / -name librxtxSerial.so 2>/dev/null  
/usr/lib/jni/librxtxSerial.so
```

Ersetzen Sie nun alle Vorkommen von **__JNI_PATH__** durch den Pfad Anteil der Ausgabe.

In unserem Beispiel also durch /usr/lib/jni

Zuguterletzt muss das System noch wissen, unter welchem User die Messung gestartet werden muss.

Ersetzen Sie hierzu alle Vorkommen von **__USER__** in der Start-Datei durch den User, unter dem Sie die Software einrichten, in unserem Fall „dfld“.

Die Startzeile einer Messung sieht im Beispiel also so aus:

```
/bin/su -l dfld -c "cd $homepath; $javapath/bin/java' -Djava.library.path=/usr/lib/jni -Xms32m
```

Zum Schluss müssen Sie nur noch als root /etc/init.d/zelle als Systemdienst ausführbar machen.

```
sudo chmod 750 /etc/init.d/dfld
```

```
sudo inserv --default dfld
```

4. Erster Start

Nun können Sie über

sudo /etc/init.d/dfld start

Ihre Messtation in Betrieb nehmen.

Im konfigurierten Log-Verzeichnis können Sie die Meldungen des Programms verfolgen.

Gehen Sie mit „cd /home/dfld/zelle/log“ in das Verzeichnis der Logdateien. Suchen Sie dort mit „ls -lrt“ nach der neuesten Datei (die unterste ist es)

Geben Sie nun tail -f dateiname an um der Datei zu folgen. Eventuell auftretende Fehler werden hier gelistet. Ein einmaliger Fehler täglich gegen Mitternacht ist normal, da hierbei auf das neue www/wwx File umgeschaltet wrd, das noch nicht existiert. Das Log sollte etwa so aussehen:

```
pi@raspberrypi ~/dfld $ cat zelle.log
LOG: Level 2
30.06.2014 20:59:56 INI: Station.ini
30.06.2014 20:59:56 Station: 999/001
30.06.2014 20:59:56 INI: Zelle.ini
30.06.2014 20:59:56 Station: RRR/SSS
30.06.2014 20:59:56 MESSTYP: 1.0 sec
30.06.2014 20:59:56 Verzeichnis /home/pi/dfld/messung/ OK
30.06.2014 20:59:56 Verzeichnis /home/pi/dfld/messung/001 neu angelegt
30.06.2014 20:59:56 Verzeichnis /home/pi/dfld/messung/RRR/SSS/ neu angelegt
30.06.2014 20:59:56 Verzeichnis /home/pi/dfld/messung/ftp/ OK
30.06.2014 20:59:56 Verzeichnis /home/pi/dfld/ OK
30.06.2014 20:59:56 Verzeichnis /home/pi/dfld/log/ OK
30.06.2014 20:59:56 Mode=NORMAL
30.06.2014 20:59:56 INI: Station.ini
30.06.2014 20:59:56 INI: Zelle.ini
30.06.2014 20:59:57 DFLD Laermmessung(Java)Version 3.0 Build-16-JUN-2014 @ Station 001/049 an /dev/ttyUSB0
[OS: LINUX64]
30.06.2014 20:59:57 Mode: Measurement and Datatransfer
30.06.2014 20:59:57 Measurement type: expected measurement period 1,0 seconds (this means 3600 measurements
per hour)
30.06.2014 20:59:57 Com port /dev/ttyUSB0 available
30.06.2014 20:59:58 Error writing status file /home/pi/dfld/Status_Zelle.dat
/home/pi/dfld/Status_Zelle.dat (No such file or directory)
java.io.FileNotFoundException: /home/pi/dfld/Status_Zelle.dat (No such file or directory)
    at java.io.FileInputStream.open(Native Method)
```

```

at java.io.FileInputStream.<init>(FileInputStream.java:146)
at java.io.FileInputStream.<init>(FileInputStream.java:101)
at dfld.zelle.zelle.IOZelle.getStatus(IOZelle.java:719)
at dfld.zelle.zelle.Batch_Zelle.getStatus(Batch_Zelle.java:358)
at dfld.zelle.zelle.ComReader_Zelle.start(ComReader_Zelle.java:267)
at dfld.zelle.zelle.Batch_Zelle.startMessung(Batch_Zelle.java:281)
at dfld.zelle.zelle.Batch_Zelle.<init>(Batch_Zelle.java:109)
at dfld.zelle.zelle.Batch_Zelle.<init>(Batch_Zelle.java:62)
at dfld.zelle.Zelle.main(Zelle.java:126)
30.06.2014 20:59:58 Start FTP-Copy
30.06.2014 20:59:58 Start measurement: 30.06.2014 / 20:59:58
30.06.2014 20:59:58 Port /dev/ttyUSB0 opened

```

Nach einer Stunde wird das erste Datenpaket auf www.dfld.de hochgeladen und kann dort angezeigt werden.

5. Auflistung der Schnittstellen

Falls unklar ist, an welcher Schnittstelle das Mikrofon angeschlossen wurde, hilft TestZelle. Dieses kleine Programm versucht 10 Sekunden von jeder seriellen Schnittstelle des Systems zu lesen und zeigt die gelesenen Werte an. Beispiel:

```

$ cd /home/dfld/zelle
$ java -classpath lib/dfld-messung.jar:/usr/share/java/RXTXcomm.jar dfld.zelle.TestZelle language en
23.02.2011 22:52:55 -----
23.02.2011 22:52:55 TestZelle 2.0/Build-23-Feb-2011
23.02.2011 22:52:55 -----
Experimental: JNI_OnLoad called.
Stable Library
=====
Native lib Version = RXTX-2.1-7
Java lib Version   = RXTX-2.1-7
23.02.2011 22:52:55 trying Port /dev/ttyS1
23.02.2011 22:52:55 Port /dev/ttyS1 opened
23.02.2011 22:52:55 /dev/ttyS1> -109 = 48.5 dB(A)
23.02.2011 22:52:56 /dev/ttyS1> -109 = 48.5 dB(A)
23.02.2011 22:52:56 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:52:57 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:52:58 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:52:58 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:52:59 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:52:59 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:00 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:01 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:01 /dev/ttyS1> -109 = 48.5 dB(A)
23.02.2011 22:53:02 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:02 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:03 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:04 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:04 /dev/ttyS1> -109 = 48.5 dB(A)
23.02.2011 22:53:05 /dev/ttyS1> -110 = 48.0 dB(A)
23.02.2011 22:53:05 trying Port /dev/ttyS0
23.02.2011 22:53:05 Port /dev/ttyS0 opened
23.02.2011 22:53:15 List of interfaces
23.02.2011 22:53:15 Name: /dev/ttyS1 open: OK read: 17 error: no, Exception: No
23.02.2011 22:53:15 Name: /dev/ttyS0 open: OK read: 0 error: no, Exception: No

```

Auf dem getesteten System hängt der Lärm-Monitor also an /dev/ttyS1(grün), von /dev/ttyS0 (rot) konnten keine Werte gelesen werden. Es wurden in 10 Sekunden 17 Werte gelesen, die Hardware läuft also im (alten) 0,6 Sekunden Takt.

6. Release Notes

Zelle 3.02 (Eric Müller, September 2014)

- Änderungen beim Timing des Ansprechens der seriellen Schnittstelle
- Messmodus des db-Online-Moduls wird nur noch temporär gesetzt
- Experimentelle Implementierung eines SIO-Recovery, falls konfigurierte serielle Schnittstelle nicht mit 9600 baud geöffnet werden kann
- Dokumentationsfehler behoben

Zelle 3.0 (Eric Müller, Juni 2014)

- Neues Installer-Script auf Shell-Basis
- Live-Datenübertragung mit 1/sec per UDP zum DFLD Server
- Möglichkeit, nachkalibrierte Mikrofone einzusetzen
- Anpassungen für Java 7
- Anpassungen für Raspberry Pi Support

Zelle 2.0b (Eric Müller, November 2011)

- Dokumentationsfehler behoben.
- Timeout beim Lesen von Seriell jetzt auf 1200ms
- Auslesen des Hardware-Modus von db-Online
- MssZeit=1 ist jetzt Default, falls nicht angegeben
- Verbesserung im Logging, die ersten 10 Messwerte werden ins Log geschrieben
-

Zelle 2.0 (Eric Müller, Januar 2011)

- Starke Vereinfachung der Software
- Online Modus mit grafischem Frontend und Auswertungsmodul entfällt
- Zeitsynchronisation erfolgt nur noch über ntpd
- Statt einer direkten DSL Unterstützung können nun eigene Scripts zum optionalen Verbindungs-Auf- und Abbau aufgerufen werden.
- Installer entfällt
- Keine Abhängigkeiten mehr zu Scripts außer dem RC-Script – Reine Java Lösung inklusive des asynchronen FTP der Daten zum Server.
- Nur noch Station.ini für lokale Konfigurationen und Zelle.ini für die DFLD Transferdaten-Konfiguration. Die Datei FtpTransfer.ini entfällt.
- Unterstützung des neuen 1-Sekunden Mess-Takts und der daraus folgenden neuen Dateistruktur
- Automatisches Anlegen der benötigten Unter-Verzeichnisstruktur unter mesung und ftp.
- RTS und DTR Leitung jetzt definiert down
- Test-Modus zur seriellen Port-Bestimmung implementiert (TestZelle)

Zelle 1.1 (Bernd Gimlott, 2005)

- Die Zelle merkt sich ihren Aufsetzpunkt. Dies bedeutet, dass bei einem Restart keine nennenswerten Werte verloren gehen.

- Die Software für die Zelle wurde von der Auswertungssoftware getrennt, welche speziell für den Berliner Raum konzipiert wurde. Dadurch wurde die Auslieferung wesentlich kleiner
- Die Anzeige wurde erweitert und mit einer Zeitachse versehen.
- Neben dem Befehl ntpimeset zur Zeitsynchronisation ist der Aufruf eines Restarts des NTP-Dienstes vorgesehen.
- Für DSL-Anwender wurde eine eigene FTP-Copy-Shell erstellt, welche den Auf- und Abbau der Verbindung unterdrückt.
- Probleme mit der Zeitsynchronisation bei Zeitkorrekturen in die Vergangenheit behoben (trat selten auf)
- Problem der Parameterübergabe bei DSL behoben
- Fehler in der Installationsprozedur behoben

•
Zelle 1.0 (Bernd Gimlott, 2004)