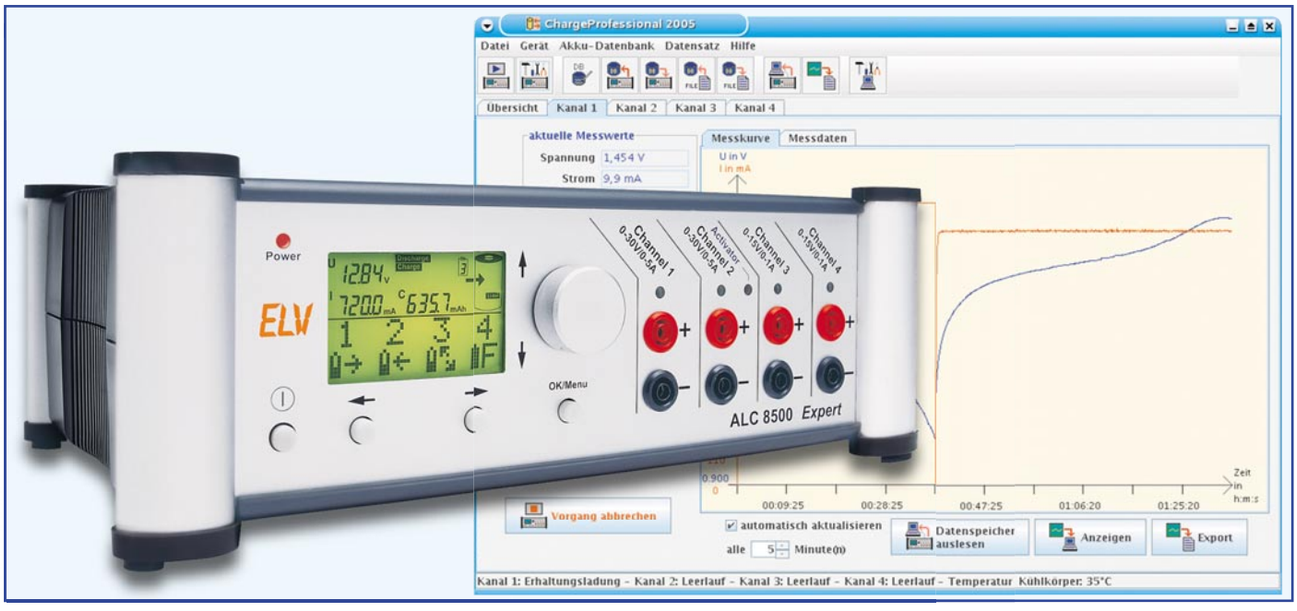

Akku-Lade-Center Teil 8

ALC 8000/ALC 8500 Expert

Linux ist in aller Munde, und wer sein ALC 8500 Expert im Bastelkeller an einem Zweitrechner betreibt, ist nicht immer bereit, sich noch eine Windows-Lizenz zu kaufen. Was liegt näher, als ein kostenloses Linux zu installieren. Mit diesem Artikel sind Sie in der Lage, ein Linux so zu konfigurieren, dass ein ALC 8500 Expert problemlos auch unter Linux mit einer Software fernsteuerbar ist. Die Beschreibungen sind sehr allgemein gehalten, so dass die Anleitung mit den verschiedensten Distributionen von Linux nachvollziehbar ist. Die Bildschirmfotos sind hier am Beispiel der Linux-Distribution Suse 10 64-Bit angefertigt.

Schrittweise zum Erfolg

Im ersten Schritt wird die Unterstützung für USB, die übrigens auch für viele andere Geräte von ELV funktioniert, eingerichtet. Anschließend wird die Laufzeitumgebung von Java angepasst bzw. installiert, falls sie noch nicht vorhanden ist. Damit die Java-Laufzeitumgebung überhaupt auf eine serielle Schnittstelle zugreifen kann, wird noch eine Erweiterung installiert. Ein kleines Testprogramm zeigt den Erfolg an, wenn die Konfigurationsarbeiten bis zu diesem Schritt korrekt ausgeführt wurden. Danach ist ChargeProfessional bereits startklar. Einige der nachfolgend verwendeten Dateien finden Sie kostenlos auf der Homepage von ELV zum Downloaden unter dem Link: http://www.elv-downloads.de/downloads/alc8xxx/alc8xxx.htm.

Treiber für USB

Dieser Treiber stellt unter dem Betriebssystem Linux für das Akku-Ladegerät ALC 8500 Expert eine virtuelle serielle Schnittstelle bereit. Die Schnittstelle trägt zumeist den Namen `ttyUSB0` bzw. wird

ftdi_sio.h [Geändert] - KWrite

Datei Bearbeiten Ansicht Lesezeichen Extras Einstellungen Hilfe

```
146  * ELV USB devices submitted by Christian Abt of ELV (www.elv.de).
147  * All of these devices use FTDI's vendor ID (0x0403).
148  *
149  * The previously included PID for the UO 100 module was incorrect.
150  * In fact, that PID was for ELV's UR 100 USB-RS232 converter (0xFB58).
151  *
152  * Armin Laeuger originally sent the PID for the UM 100 module.
153  */
154 #define FTDI_ELV_UR100_PID      0xFB58  /* USB-RS232-Umsetzer (UR 100) */
155 #define FTDI_ELV_UM100_PID      0xFB5A  /* USB-Modul UM 100 */
156 #define FTDI_ELV_UO100_PID      0xFB5B  /* USB-Modul UO 100 */
157 #define FTDI_ELV_ALC8500_PID    0xF06E  /* ALC 8500 Expert */
158 /* Additional ELV PIDs that default to using the FTDI D2XX drivers on
159  * MS Windows, rather than the FTDI Virtual Com Port drivers.
160  * Maybe these will be easier to use with the libftdi/libusb user-space
161  * drivers, or possibly the Comedi drivers in some cases. */
162 #define FTDI_ELV_CLI7000_PID    0xFB59  /* Computer-Light-Interface (CLI 7000) */
163 #define FTDI_ELV_PPS7330_PID    0xFB5C  /* Processor-Power-Supply (PPS 7330) */
164 #define FTDI_ELV_TFM100_PID     0xFB5D  /* Temperartur-Feuchte Messgeraet (TFM 100) */
165 #define FTDI_ELV_UDF77_PID      0xFB5E  /* USB DCF Funkurh (UDF 77) */
166 #define FTDI_ELV_UIO88_PID      0xFB5F  /* USB-I/O Interface (UIO 88) */
167 #define FTDI_ELV_UAD8_PID       0xF068  /* USB-AD-Wandler (UAD 8) */
168 #define FTDI_ELV_UDA7_PID       0xF069  /* USB-DA-Wandler (UDA 7) */
169 #define FTDI_ELV_USI2_PID       0xF06A  /* USB-Schrittmotoren-Interface (USI 2) */
170 #define FTDI_ELV_T1100_PID      0xF06B  /* Thermometer (T 1100) */
171 #define FTDI_ELV_PCD200_PID     0xF06C  /* PC-Datenlogger (PCD 200) */
172 #define FTDI_ELV_ULA200_PID     0xF06D  /* USB-LCD-Ansteuerung (ULA 200) */
173 #define FTDI_ELV_FHZ1000PC_PID  0xF06F  /* FHZ 1000 PC */
174 #define FTDI_ELV_CSI8_PID       0xE0F0  /* Computer-Schalt-Interface (CSI 8) */
175 #define FTDI_ELV_EM1000DL_PID   0xE0F1  /* PC-Datenlogger fuer Energiemonitor (EM 1000
    DL) */
176 #define FTDI_ELV_PCK100_PID     0xE0F2  /* PC-Kabeltester (PCK 100) */
177 #define FTDI_ELV_RFP500_PID     0xE0F3  /* HF-Leistungsmesser (RFP 500) */
178 #define FTDI_ELV_FS20SIG_PID    0xE0F4  /* Signalgeber (FS 20 SIG) */
179 #define FTDI_ELV_WS300PC_PID    0xE0F6  /* PC-Wetterstation (WS 300 PC) */
180 #define FTDI_ELV_FHZ1300PC_PID  0xE0E8  /* FHZ 1300 PC */
181 #define FTDI_ELV_WS500_PID      0xE0E9  /* PC-Wetterstation (WS 500) */
182
183 /*
```

Bild 1: Auszug aus dem Quelltext für den USB-Treiber

bei mehreren Geräten einfach eine fortlaufende Nummer angehängt (`ttyUSB1`, `ttyUSB2` usw.). Wer bereits den Kernel 2.6.13 (z. B. in Suse 10 enthalten) oder höher installiert hat, kann direkt zum nächsten Schritt übergehen, da die notwendige Unterstützung ab dieser Version des Kernels bereits integriert ist. Falls der bestehende Kernel nicht vollständig durch eine neue Version ersetzt werden soll, reicht es aus, zwei Dateien im Quelltext durch eine andere Version auszutauschen und den bestehenden Kernel neu zu kompilieren. Die beiden Dateien sind üblicherweise abgelegt in diesem Verzeichnisbaum: `/usr/src/linux-2.6.13-15/drivers/usb/serial`. Die Bezeichnung `linux-2.6.13-15` ist selbstverständlich durch den Namen Ihrer Version zu ersetzen. Die beiden Dateien tragen die Namen `ftdi_sio.c` und `ftdi_sio.h` und enthalten die Version 1.4.3 des Treibers. Wenn Sie sich etwas mehr auskennen, werfen Sie einen genaueren Blick in die Datei `ftdi_sio.h` ab Zeile 146, wie in Abbildung 1 gezeigt. Für die Geräte UR 100 (USB-RS232-Umsetzer), UO 100 und UM 100 ist die Unterstützung standardmäßig aktiviert. In den Zeilen 162 bis 181 finden Sie weitere Geräte von ELV; falls Sie diesen Geräten auch einen virtuellen seriellen Port zuweisen möchten, kommentieren Sie die jeweiligen Zeilen in der Datei `ftdi_sio.c` ab Zeile 420 einfach aus.

Der Erfolg der Konfiguration lässt sich leicht feststellen. Stecken Sie einfach das USB-Kabel des ALC 8500 Expert in einen freien Anschluss am Computer und geben den Befehl `dmesg` in einer Konsole ein. Die letzten Meldungen des Kernels werden angezeigt. Das Bildschirmfoto Abbildung 2 zeigt eine erfolgreiche Einbindung des Treibers. In diesem Fall wurde dem Gerät die Schnittstelle `ttyUSB0` zugewiesen. Eine weitere Möglichkeit bietet das Programm `usbview` (siehe Abbildung 3). Dieses Programm zeigt alle angeschlossenen Geräte für den USB an. Geräte, welche nicht über einen passenden Treiber verfügen, werden rot dargestellt. Wenn das ALC 8500 Expert dort in schwarzer Schrift erscheint, hat alles geklappt. Das Programm `usbview` ist unter der GNU-Lizenz frei erhältlich und in vielen Distributionen von Linux bereits integriert.

Java einrichten …

Leider unterscheidet sich die Handhabung der Java-Umgebung je nach Distribution sehr stark. Zudem ist noch eine Erweiterung nachträglich zu installieren, auch hierfür unterscheidet sich leider die Vorgehensweise sehr. Glücklicherweise kann sich jeder Benutzer selbst eine Variante in

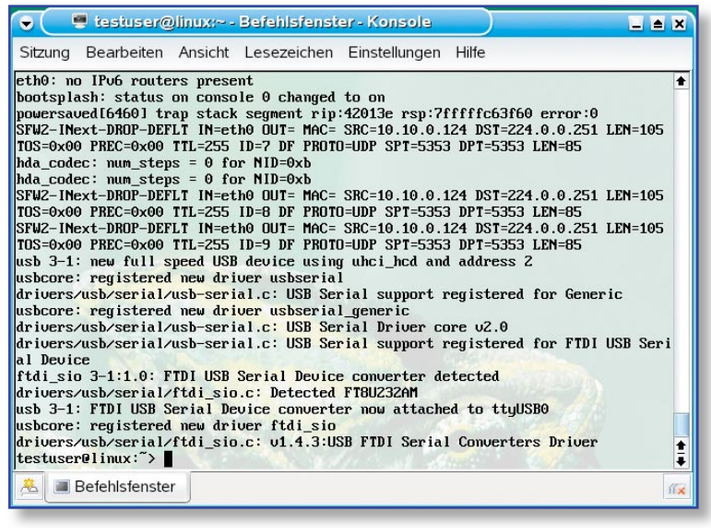

Bild 2: Der Treiber für USB wurde erfolgreich geladen, das ALC 8500 Expert hat jetzt die virtuelle serielle Schnittstelle „ttyUSB0“.

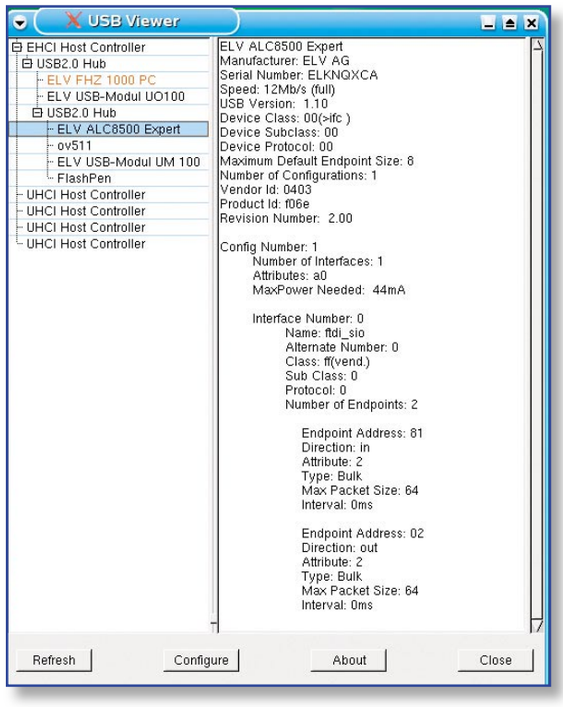

Bild 3: Das Programm „usbview“ zeigt alle erkannten Geräte in einer Baumstruktur an.

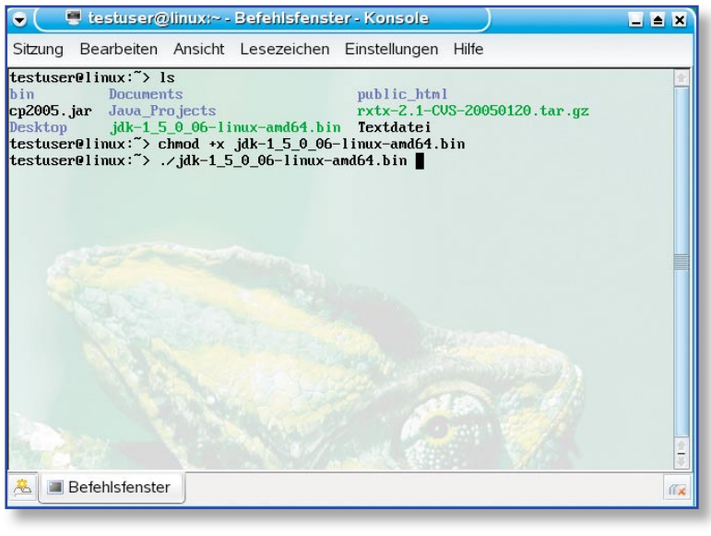

Bild 4: Die Java-Entwicklungsumgebung wird installiert.

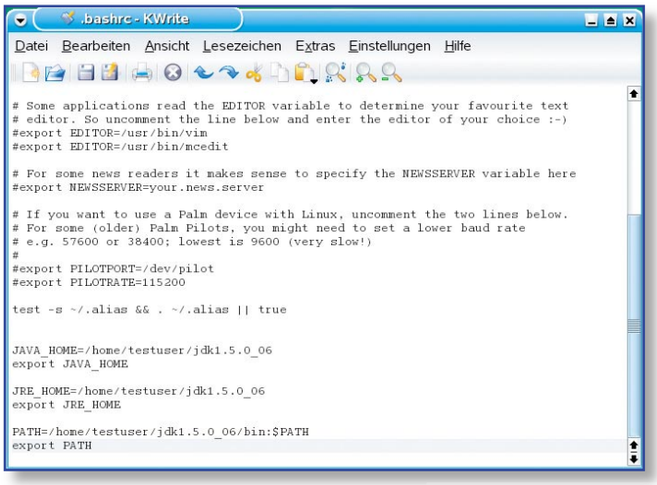

Bild 5: Die Systemvariablen werden an die Installation angepasst.

seinem Home-Verzeichnis installieren. Eine Entwicklungsumgebung der JVM (Java Virtual Machine) kann kostenlos auf der Internet-Seite `java.sun.com` bezogen werden. Dabei ist nicht die rpm-Variante, sondern einfach das Binärfile herunterzuladen. In unserem Beispiel haben wir die Datei `jdk-1_5_0_06-linux-amd64.bin` verwendet. Nach dem Download werden mit folgendem Befehl die Rechte so gesetzt, dass die Datei ausführbar ist: `chmod +x jdk-1_5_0_06-linux-amd64.bin`. Schließlich wird mit dem Befehl `./jdk-1_5_0_06-linux-amd64.bin` die Installation gestartet.Abbildung 4 zeigt, wie es geht.

... und konfigurieren

Drei Systemvariablen müssen jetzt noch auf die frisch installierte Version angepasst werden: `JAVA_HOME`, `JRE_HOME` und `PATH`. Wir gehen hier davon aus, dass Java in dem Pfad `/home/testuser/jdk1.5.0_06` installiert ist. Entweder sind die nachfolgenden Befehle nach jedem Systemstart einzugeben oder die folgenden Zeilen werden zu der `bash.rc` hinzugefügt:

```
1 JAVA_HOME=/home/
  testuser/jdk1.5.0_06
2 export JAVA_HOME
3 JRE_HOME=/home/
  testuser/jdk1.5.0_06
4 export JRE_HOME
5 PATH=/home/testuser/
  jdk1.5.0_06/bin:$PATH
6 export PATH
```

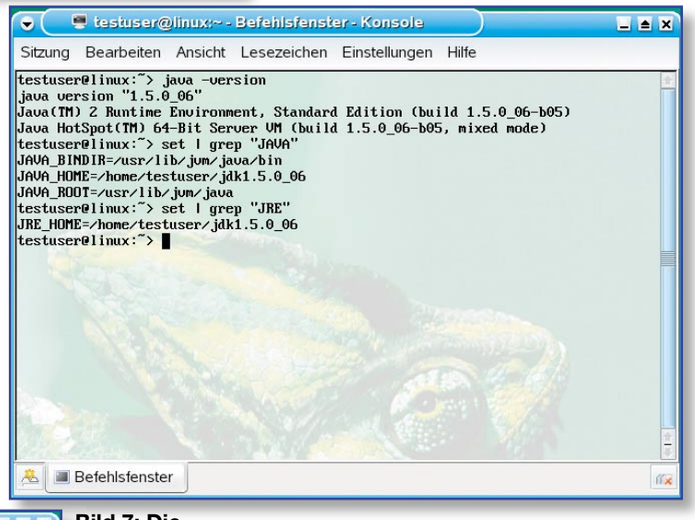

Bild 6: Die Konfiguration der Java-Entwicklungsumgebung wird überprüft.

Der Pfad `/home/testuser/jdk1.5.0_06` ist in jedem Fall an Ihre Installation anzupassen. Die Abbildung 5 zeigt, wie die `bash.rc` aussehen kann.

Der Befehl `java -version` auf einer Konsole zeigt die Version der „aktiven" Java-Laufzeitumgebung an. Wenn diese der soeben installierten Version entspricht, war die Installation erfolgreich. Mit dem Befehl `set | grep "JAVA"` und `set | grep "JRE"` lässt sich jeweils prüfen, ob die Systemvariablen passend gesetzt sind. Die Abbildung 6 zeigt einen erfolgreichen Test.

Java und Hardware

Leider bietet Java standardmäßig keinen Zugriff auf serielle Schnittstellen, da diese Programmiersprache auf Plattformunabhängigkeit ausgelegt ist. Es gibt jedoch einige Erweiterungen, die diesen Zugriff erlauben. Eine solche Erweiterung ist RXTX, dieses Paket ist über die GNU-Lizenz kostenlos erhältlich auf der Internet-Seite `www.rxtx.org`. Im Downloadbereich dieser Seite sind auch die zugehörigen Quelltexte erhältlich, nachfolgend beschrieben ist die Variante RXTX 2.1, die Datei heißt dort `rxtx-2.1-CVS-20050120.tar.gz`.

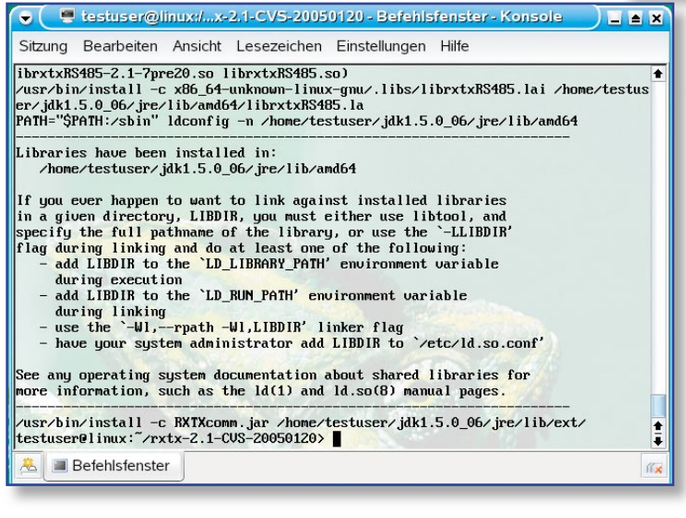

Bild 7: Die Erweiterung RXTX wurde erfolgreich kompiliert.

Dieses Archiv wird zunächst in ein neues Verzeichnis ausgepackt. Eine detaillierte Anleitung zum Einrichten enthält die Datei `INSTALL` des Archivs. Damit immer nur ein Prozess gleichzeitig auf eine serielle Schnittstelle zugreifen kann, gibt es den so genannten lock-daemon. Dazu muss der angemeldete Benutzer die entsprechenden Rechte besitzen, in diesem Fall muss er Mitglied der Gruppe `UUCP` sein. Falls der angemeldete Benutzer weder Mitglied dieser Gruppe ist noch hinzugefügt werden kann, installiert man die Erweiterung ohne den lock-daemon. Allerdings wird dann nicht überwacht, ob zwei Prozesse gleichzeitig auf eine Schnittstelle zugreifen. Diese Vorgehensweise wird nicht empfoh-

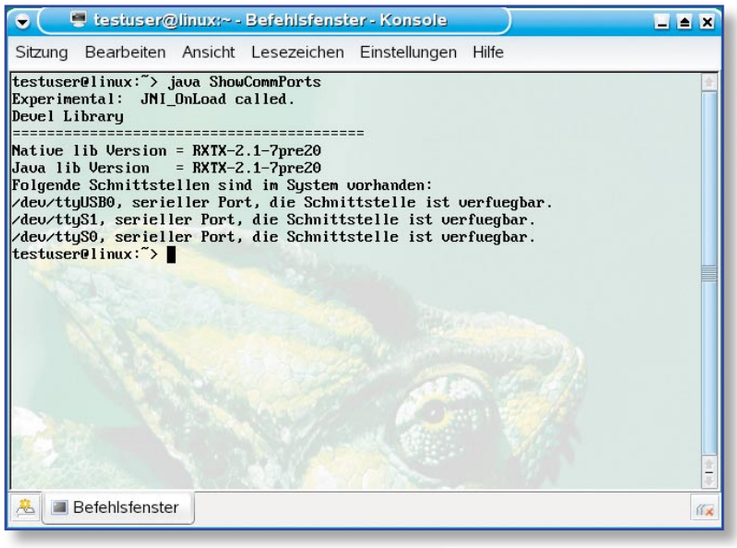

Bild 8: Ein Aufruf des Testprogramms „ShowCommPorts“

von ELV ist die Datei `ShowCommPorts.java`. Kompiliert wird dieses Programm mit dem Befehl `javac ShowCommPorts.java`. Wenn jetzt keine Fehlermeldungen erscheinen, war das Kompilieren erfolgreich. Starten Sie das Programm mit dem Befehl `java ShowCommPorts`. Das Programm gibt alle verfügbaren seriellen Schnittstellen aus. Falls das ALC 8500 Expert ebenfalls angeschlossen ist, wird es (meist) als `ttyUSB0` angezeigt. In der Abbildung 8 sehen Sie ebenfalls einen erfolgreichen Programmlauf. Die Installation war erfolgreich!

… und jetzt geht es los

Ebenfalls im Downloadbereich von ELV zum ALC 8500 Expert gibt es die Linux-Variante von ChargeProfessional. Nach dem Aus-

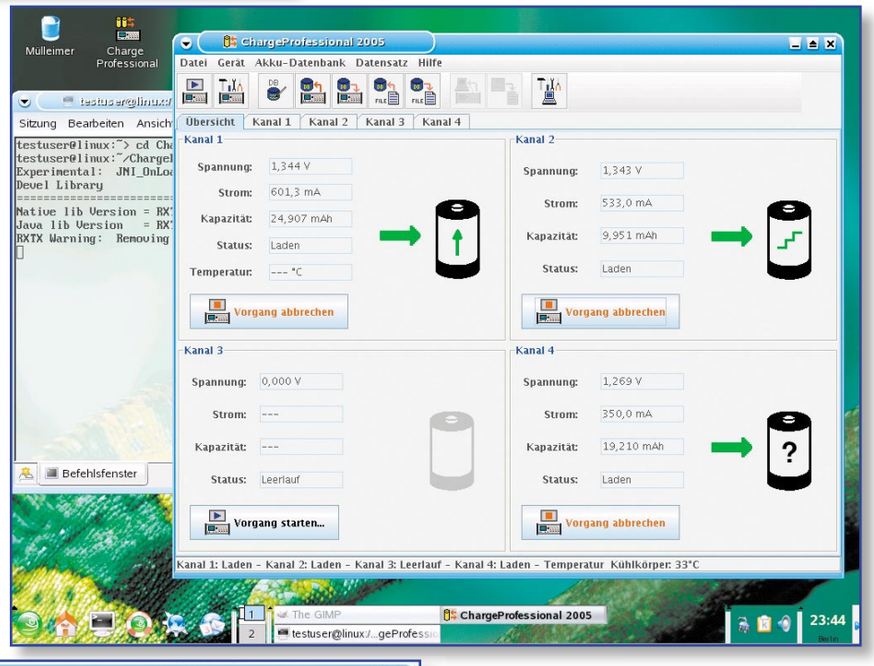

Bild 9: Das Hauptfenster von ChargeProfessional – hier in der Linux-Variante

len. Mit dem Befehl `./configure` bzw. `./configure --disable-lockfiles` für die Installation ohne lock-daemon, wird die Konfiguration vorbereitet. Mit einem abschließenden `make install` wird die Erweiterung erstellt und die erzeugten Dateien automatisch in die entsprechenden Verzeichnisse der Java-Laufzeitumgebung kopiert. In der Abbildung 7 ist zu sehen, wie ein erfolgreicher Kompiliervorgang aussieht. Damit dieser Schritt funktioniert, müssen die Pfade – wie bei der Installation der Java-Umgebung beschrieben – korrekt in den Systemvariablen eingetragen sein.

Noch ein Test …

Ebenfalls im Archiv aus dem Download

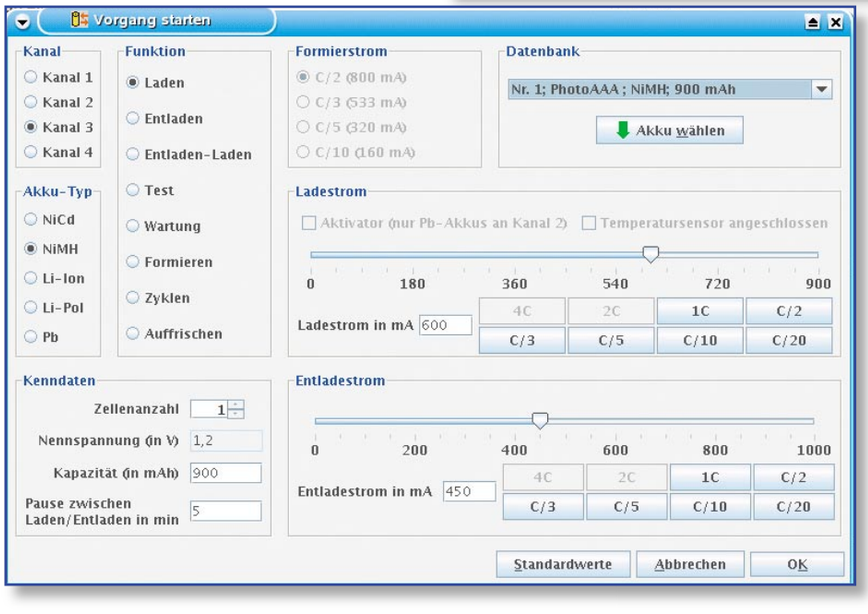

Bild 10: Starten eines Ladevorgangs in der Linux-Variante von ChargeProfessional

packen des Archivs wird das Programm mit dem Befehl `java -jar cp2005_v160.jar` gestartet. Selbstverständlich können Sie sich hierzu auch eine Verknüpfung auf dem Desktop anlegen. Die Abbildungen 9 und 10 zeigen, wie ChargeProfessional auch unter Linux das ALC 8500 Expert bedient. Bei Problemen sind die Dateien `error.txt` und `log.txt` hilfreich, die im Programmverzeichnis von ChargeProfessional automatisch angelegt werden.

Mit dieser Anleitung schließt die Artikelserie rund um das ALC 8500 Expert. Damit haben Sie einen umfassenden Überblick über die Interna des ALC erhalten, kennen das Übertragungsprotokoll und können es jetzt sogar unter Linux verwenden. ELV