
AXbo Schlafphasenwecker und Linux

By rac

Published: 28.12.2008 - 18:33

!!! UPDATE !!!

Laut einem Kommentar von Jens Ougaard (aXbo Support?) auf dem Blog von crox kann man nun unter http://www.axbo.com/webstart/aXbo_unix_2_0_14.sh eine Unix taugliche Version bekommen.

 **SCHLAFPHASENWECKER**



Dank dem Blogeintrag von [crox](#) kann man die aXbo research Software in wenigen schritten unter Linux zum laufen bringen.

Die Update funktion oder die direkte Druckfunktion laufen mit dieser Anleitung nicht, auf diese kann man jedoch auch verzichten obwohl es schon schön wäre wenn die Östereicher die install4j packages gleich auch für unix auf die Seite laden würden (sollte ja mit install4j kein Problem sein).

Der [USB 2 UART chip von CYGNAL](#) den sie verwenden, läuft bei meinem Ubuntu 8.10 Out of the Box und meldet sich mit:

```
[13361.518104] usb 1-2: new full speed USB device using uhci_hcd and address 4
[13361.680967] usb 1-2: configuration #1 chosen from 1 choice
[13361.685360] cp2101 1-2:1.0: cp2101 converter detected
[13361.797683] usb 1-2: reset full speed USB device using uhci_hcd and address 4
[13361.947598] usb 1-2: cp2101 converter now attached to ttyUSB0
```

Die Software muss wie von crox beschrieben auf der Webseite von aXbo unter downloads heruntergeladen (registrierung mit Seriennummer notwendig) und mit Wine installiert werden (nur die applikation ohne die Treiber).

Leider kann man zurzeit nur die Version aXbo_windows_2_0_11.exe herunterladen, die nach dem ersten Starten gleich mit der Updatefunktion kommt und zum Herunterladen der Version 2.0.13 einlädt.

Bei mir hatte Wine bzw crossover office ein erhebliches Darstellungsproblem und stürzte auch immer beim Suchen nach dem Wecker ab.

Deshalb starten wir das Java Programm nativ unter Linux.

Zuerst installieren wir die [RXTX library](#) unter ubuntu mit

```
sudo aptitude install librxtx-java
```

oder mit der methode auf dem Blogeintrag von crox.

Erstellt einen Applikations root (z.B. "mkdir ~/axbo") und kopiert den lib folder aus dem wine c-drive rein

(z.B. "cp -rp ~/.cxoffice/axbo/drive_c/Program Files/aXbo/lib/ ~/axbo" oder

wenn ihr wine benutzt mit "cp -rp ~/.wine/drive_c/Program Files/aXbo/lib ~/aXbo").

Wechselt in Applikationsroot (z.B. "cd ~/axbo") und verschiebt die datei axbo.jar aus dem lib folder hierhin (z.B. "mv lib/axbo.jar .").

Zum starten der applikation könnt ihr noch ein starterscript erstellen:

```
echo "java -Djava.library.path=/usr/lib/ -jar axbo.jar" > axbo.sh  
chmod +x axbo.sh
```

Das research Programm kann nun mit "./axbo.sh" ausgeführt werden.

Die Exceptions die von den install4j library geworfen werden könnt ihr getrost ignorieren (Updatefunktion geht natürlich nicht mehr).

```
$ ./axbo.sh  
Exception in thread "main" java.lang.NoClassDefFoundError:  
com/install4j/api/update/UpdateScheduleRegistry  
    at com.dreikraft.infactory.Axbo.init(Axbo.java:229)  
    at com.dreikraft.infactory.Axbo.main(Axbo.java:125)
```

Caused by: java.lang.ClassNotFoundException: com.install4j.api.update.UpdateScheduleRegistry
at java.net.URLClassLoader\$1.run(URLClassLoader.java:200)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader.java:188)
at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
at sun.misc.Launcher\$AppClassLoader.loadClass(Launcher.java:276)
at java.lang.ClassLoader.loadClass(ClassLoader.java:251)
at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:319)
... 2 more

Wenn ihr den aXbo angeschlossen habt könnt ihr nun mit "aXbo finden" den Wecker suchen und die Daten herunterladen bzw. Sounds Raufladen.

Beispielausgabe auf der Konsole beim herunterladen der Schlafdaten:

```
28.12.2008 19:12:41 com.dreikraft.infactory.controller.AxboFrameController testPort
INFO: testing system comm port: /dev/ttyUSB0
Experimental: JNI_OnLoad called.
Stable Library
=====
Native lib Version = RXTX-2.1-7
Java lib Version = RXTX-2.1-7
RXTX Warning: Removing stale lock file. /var/lock/LCK..ttyUSB0
28.12.2008 19:12:41 com.dreikraft.infactory.data.AxboCommandUtil getCheckCmd
INFO: reading status
28.12.2008 19:13:45 com.dreikraft.infactory.controller.AxboFrameController storeClockData
INFO: start reading clock data
28.12.2008 19:13:45 com.dreikraft.infactory.data.AxboCommandUtil getLogDataCmd
INFO: retrieving log data
```

Beim direkten Drucken aus dem Programm gibt es bei mir folgenden Fehler:

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException: null attribute
at sun.print.IPPPrintService.isAttributeValueSupported(IPPPrintService.java:1147)
at sun.print.ServiceDialog$OrientationPanel.updateInfo(ServiceDialog.java:2121)
at sun.print.ServiceDialog$PageSetupPanel.updateInfo(ServiceDialog.java:1263)
at sun.print.ServiceDialog.initPageDialog(ServiceDialog.java:266)
at sun.print.ServiceDialog.(ServiceDialog.java:243)
at sun.print.RasterPrinterJob.pageDialog(RasterPrinterJob.java:657)
```

```
at sun.print.RasterPrinterJob.pageDialog(RasterPrinterJob.java:614)
at org.jfree.chart.ChartPanel.createChartPrintJob(ChartPanel.java:2133)
at com.dreikraft.infactory.gui.DataFrame.print(DataFrame.java:812)
at com.dreikraft.infactory.gui.DataFrame.btnPrintActionPerformed(DataFrame.java:610)
at com.dreikraft.infactory.gui.DataFrame.access$500(DataFrame.java:49)
at com.dreikraft.infactory.gui.DataFrame$6.actionPerformed(DataFrame.java:254)
at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:1995)
at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2318)
at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:387)
at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:242)
at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:236)
at java.awt.AWTEventMulticaster.mouseReleased(AWTEventMulticaster.java:272)
at java.awt.Component.processMouseEvent(Component.java:6041)
at javax.swing.JComponent.processMouseEvent(JComponent.java:3265)
at java.awt.Component.processEvent(Component.java:5806)
at java.awt.Container.processEvent(Container.java:2058)
at java.awt.Component.dispatchEventImpl(Component.java:4413)
at java.awt.Container.dispatchEventImpl(Container.java:2116)
at java.awt.Component.dispatchEvent(Component.java:4243)
at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4322)
at java.awt.LightweightDispatcher.processMouseEvent(Container.java:3986)
at java.awt.LightweightDispatcher.dispatchEvent(Container.java:3916)
at java.awt.Container.dispatchEventImpl(Container.java:2102)
at java.awt.Window.dispatchEventImpl(Window.java:2440)
at java.awt.Component.dispatchEvent(Component.java:4243)
at java.awt.EventQueue.dispatchEvent(EventQueue.java:599)
at java.awt.EventQueue.dispatchEvent(EventQueue.java:599)
at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:273)
at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:183)
at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:173)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:168)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:160)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:121)
```

Man kann jedoch das Diagramm als PNG exportieren und das PNG file ausdrucken.

Related Links

- [Herstellerseite](#)
- [Source Howto Blogeintrag](#)
- [Yet Another Sleep Phase Analysis Program \(nicht kostenlos\)](#)

-->

Trackback URL for this post:

<http://www.2030.tk/trackback/213>