RXTX JAVA LIBRARY

RXTX is a Java library, using a native implementation (via JNI), providing serial and parallel communication for the Java Development Toolkit (JDK). All deliverables are under the GNU LGPL license. It is based on the specification for Sun's Java Communications API, but the package name is gnu.io.

With the Communications API you can obtain a set of objects representing the RS232 serial ports and IEEE 1284 parallel ports on a platform. The serial and parallel port classes provide low level I/O capabilities.

HOW TO INSTALL

1. place the .so (linux) or .dll (windows) in the LD_LIBRARY_

2. place the RXTXcomm.jar in the CLASSPATH

See for example :

- http://rxtx.qbang.org/wiki/index.php/Main_Page
- http://www.cloudhopper.com/opensource/rxtx (x64 binaries)
- http://www.agaveblue.org/howtos/Comm_How-To.shtml (linux)
- http://www.particle.kth.se/~lindsey/JavaCourse/Book/Part3/Chapter23/commAPI.html

Stappen :

- Download rxtx van: http://rxtx.qbang.org/wiki/index.php/download. Hier moet je de binary van rxtx 2.1-7r2 (stable) hebben. Maar geeft dit problemen (x64 W7), probeer dan rxtx 2.2pre2 (**prerelease**).
- Unzip de zip file en kopieer RXTXcomm.jar naar C:\Program Files\Java\jdk1.6.0_18\jre\lib\ext.
- Kopieer rxtxSerial.dll naar C:\Program Files\Java\jdk1.6.0_18\jre\bin.
- Daarnaast natuurlijk de juiste port name opgeven (afhankelijk van OS).

HOW TO SET THE CLASSPATH

On Linux the Classpath contains names seperated by a colon ":" , on Windows names are separated by a semi colon ";".

To Compile : let current dir './' contain all java files + RXTXcomm.jar

```
javac -cp .:RXTXcomm.jar Multisegment.java (Linux)
javac -cp .;RXTXcomm.jar Multisegment.java (Windows)
```

To execute : arg is com port (Linux default = /dev/ttyS0)

```
java -cp .:RXTXcomm.jar Multisegment (Linux)
java -cp .;RXTXcomm.jar Multisegment (Windows)
```

ClassNotFoundException is an Exception and will be thrown when Java dynamically tries to load a particular Class at Runtime and doesn't find that on Java classpath.

NoClassDefFoundError comes when a particular class was present in Java Classpath at compile time but is not available at run-time.

See

- Setting the class path : http://javarevisited.blogspot.com/2011/01/how-classpath-work-in-java.html
- Setting the class path : http://download.oracle.com/javase/6/docs/technotes/tools/solaris/classpath.html
- Java compiler : http://download.oracle.com/javase/6/docs/technotes/tools/solaris/javac.html