### Git clone <https://pkocsis@github.com/pkocsis/JavaFTD2XX.git>

### //////////////////////////////////////////////////////////////////////////////////////////////////

### CMD in windows

### java -jar DenkoviRelayCommandLineTool\_10.jar list (get **DEVICE SERIAL NUMBER= DAE000M5)**

### java -jar DenkoviRelayCommandLineTool\_10.jar **DAE000M5 4 1 1**

### **[DEVICE SERIAL NUMBER= DAE000M6] [RELAY COUNT=4] [RELAY NUMBER=1~4] (status** 1-on)

## How it works

To interact with the devices the plugin uses the [JavaFTD2XX](http://kenai.com/projects/javaftd2xx/pages/Home) library.

**NOTE FOR LINUX USERS**: as reported on the official site you should be sure that you have permission to write to the chip. (/dev/bus/usb/[BUS]/[DEVICE], you can determinate the exact location by lsusb). By default you will not have permission, so you must chown it every time when you physically connect, or make an UDEV rule: [this way.](https://wiki.archlinux.org/index.php/Udev#Accessing_Firmware_Programmers_and_USB_Virtual_Comm_Devices) You must fit idVendor and idProduct values to ftdi's, using lsusb. For example a device:

Bus 001 Device 006: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC

Bus 004 Device 003 begin\_of\_the\_skype\_highlighting 004 Device 003 FREE  end\_of\_the\_skype\_highlighting : ID (this is idVendor-->)0b38:0003(<--this is idProduct) ...

### Accessing firmware programmers and USB virtual comm devices

The following ruleset will allow normal users (within the "users" group) the ability to access the [USBtinyISP](http://www.ladyada.net/make/usbtinyisp/) USB programmer for AVR microcontrollers and a generic (SiLabs [CP2102](http://www.silabs.com/products/interface/usbtouart)) USB to UART adapter, the [Atmel AVR Dragon](http://www.atmel.com/tools/AVRDRAGON.aspx?tab=overview) programmer, and the [Atmel AVR ISP mkII](http://www.atmel.com/tools/AVRISPMKII.aspx). Adjust the permissions accordingly. Verified as of 31-10-2012.

/etc/udev/rules.d/50-embedded\_devices.rules

# USBtinyISP Programmer rules

SUBSYSTEMS=="usb", ATTRS{idVendor}=="1781", ATTRS{idProduct}=="0c9f", GROUP="users", MODE="0666"

SUBSYSTEMS=="usb", ATTRS{idVendor}=="16c0", ATTRS{idProduct}=="0479", GROUP="users", MODE="0666"

# USBasp Programmer rules http://www.fischl.de/usbasp/

SUBSYSTEMS=="usb", ATTRS{idVendor}=="16c0", ATTRS{idProduct}=="05dc", GROUP="users", MODE="0666"

# Mdfly.com Generic (SiLabs CP2102) 3.3v/5v USB VComm adapter

SUBSYSTEMS=="usb", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", GROUP="users", MODE="0666"

#Atmel AVR Dragon (dragon\_isp) rules

SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2107", GROUP="users", MODE="0666"

#Atmel AVR JTAGICEMKII rules

SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2103", GROUP="users", MODE="0666"

#Atmel Corp. AVR ISP mkII

SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2104", GROUP="users", MODE="0666"

### ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

### 

### MASK ==🡺 freedomotic.jfrontend

### while (!loginSuccessfull) {

### int result = JOptionPane.showConfirmDialog(

### null,

### new Object[]{nameLbl, jnf, label, jpf},

### I18n.msg( "enter\_credentials"),

### JOptionPane.OK\_CANCEL\_OPTION);

### if (result == JOptionPane.OK\_OPTION) {

### loginSuccessfull = Auth.login(jnf.getText(), jpf.getPassword());

### } else {

### loginSuccessfull = true;

### }

### }

### ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

### java -jar ./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/freedomotic.jar

### cd ./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world

### vi ./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world/

### vi ./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world/hello-world-manifest.xml

### find -name hello-world-3.0.jar

### head hello-world-manifest.xml

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/hello-world/hello-world-3.0.jar pi@192.168.1.249:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/hello-world/hello-world-3.0.jar pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/hello-world/hello-world-manifest.xml pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world

### pscp -r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/hello-world/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world

### pscp -l pi -pw raspberry –r c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/hello-world/lib/\*.\* pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world/lib

### change root password 🡺 sudo passwd root

### pscp -l root -pw lu571022 root@192.168.1.197:/etc/inittab c:/latestsetup/

### pscp -l pi -pw raspberry pi@192.168.1.198:/home/pi/.bashrc c:/latestsetup

### pscp -l root -pw lu571022 c:/latestsetup/inittab root@192.168.1.12:/etc

### pscp -l pi -pw raspberry c:/latestsetup/.bashrc pi@192.168.1.198:/home/pi

### //----------

### pscp -l root -pw lu571022 root@192.168.1.12:/etc/network/interfaces c:/latestsetup/

### pscp -l root -pw lu571022 root@10.25.187.220:/etc/wpa\_supplicant/\*.\* c:/latestsetup/wpa\_supplicant

### pscp -r -l root -pw lu571022 c:/latestsetup/wpa\_supplicant/\* root@192.168.1.12:/etc/wpa\_supplicant

### pscp -l root -pw lu571022 c:/latestsetup/interfaces root@10.25.187.53:./etc/network/

### pscp -l root -pw lu571022 c:/latestsetup/jna-4.0.0.jar root@10.25.187.224:/usr/lib/jvm/jdk-7-oracle-armhf/jre/lib

### pscp -l root -pw lu571022 c:/latestsetup/interfaces root@10.25.187.53:./etc/network/

### pscp -l pi -pw raspberry pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world/hello-world-manifest.xml c:/latestsetup/tmp

### pscp -l pi -pw raspberry c:/latestsetup/hello-world-manifest.xml pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world

### pscp -l pi -pw raspberry pi@10.25.187.224:./Desktop/LXTerminalCMD.txt c:/latestsetup/

### pscp -l pi -pw raspberry pi@10.25.187.223:./.xinitrc c:/latestsetup/

### pscp -l pi -pw raspberry c:/latestsetup/LXTerminalCMD pi@10.25.187.223:./

### pscp -l pi -pw raspberry c:/latestsetup/.xinitrc pi@10.25.187.224:./

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/freedomotic.jar pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### pscp –r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/lib/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/lib

### pscp -r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/config/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/config

### pscp -r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/config/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/config

### important!!!!! Related with data!!!!

### pscp -r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/data/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/data

### important!!!!! No related with i18n!!!!

### pscp -r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/i18n/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/i18n

### important!!!!! Related with config!!!!

### pscp -r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/config/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/config

### important!!!!! Related with config!!!!

### pscp -r -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/config/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/config

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/freedomotic.jar pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### pscp -r -l pi -pw raspberry pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/\* c:/LatestSetup/RASPI

### pscp -r -l pi -pw raspberry c:/LatestSetup/RASPI/\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic.jar pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### pscp –r -l pi -pw raspberry pi@192.168.1.12:/home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/\* c:/LatestSetup/tmp

### pscp -l pi -pw raspberry –r c:/LatestSetup/freedomotic/framework/freedomotic-core/data/resources/\*.\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/data/resources

### pscp -l pi -pw raspberry –r c:/LatestSetup/freedomotic/framework/freedomotic-core/data/furn/\*.\* pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/data/furn

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/freedomotic.jar pi@192.168.1.12:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/freedomotic.jar pi@192.168.1.12:./Desktop/freedomotic

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/lib/junit-4.11.jar pi@192.168.1.12:./Desktop/freedomotic/lib

### pscp -l pi -pw raspberry -r c:/LatestSetup/freedomotic/framework/freedomotic-core/i18n/\*.\* pi@10.25.187.223:./Desktop/freedomotic/i18n

### pscp -l pi -pw raspberry -r c:/LatestSetup/freedomotic/framework/freedomotic-core/config/\*.\* [pi@10.25.187.223:./Desktop/freedomotic/config](mailto:pi@10.25.187.87:./Desktop/freedomotic/config)

### pscp -l pi -pw raspberry -r c:/LatestSetup/freedomotic/framework/freedomotic-core/data/\*.\* pi@10.25.187.223:./Desktop/freedomotic/data

### pscp -l pi -pw raspberry -r c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/\*.\* pi@10.25.187.223:./Desktop/freedomotic/plugins

### pscp -l pi -pw raspberry –r c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/\*.\* pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### pscp -l pi -pw raspberry –r c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/frontend-java/\*.\* pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/frontend-java

### pscp -l pi -pw raspberry –r c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/frontend-java/lib/\*.\* pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/frontend-java/lib

### pscp -l pi -pw raspberry –r c:/LatestSetup/freedomotic/framework/freedomotic-core/plugins/devices/frontend-java/data/i18n/\*.\* [pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/frontend-java/data/i18n/](mailto:pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/frontend-java/data/i18n/)

### pscp -l pi -pw raspberry c:/LatestSetup/freedomotic/framework/freedomotic-core/target/freedomotic-core/freedomotic.jar pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### pscp -l pi -pw raspberry c:/latestsetup/CDM/libftd2xx1.1.12.tar pi@10.25.187.223:./

### pscp -l pi -pw raspberry c:/latestsetup/CDM/DenkoviRelayCommandLineTool\_10.jar pi@10.25.187.223:./

### pscp -l root -pw lu571022 c:/LatestSetup/raspi-IP/interfaces pi@10.25.187.223:./Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### root ------ > etc/network/interfaces

### /dev/ttyUSB0---COM4

### SinBerBEST-PC: 10.25.187.225

### Rasp1: 10.25.187.223

### Rasp1: 10.25.187.223

### Rasp1: 10.25.187.222

### Rasp1: 10.25.187.221

### ==========

### User : pi

### Java Home: /usr/lib/jvm/jdk-7-oracle-armhf/jre

### Java Library Path: {/usr/java/packages/lib/arm:/lib:/usr/lib}

### Program path: /home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a

### Java Version: 1.7.0\_40

### Resources Path: /home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0. 83-0504a4a/data/resources

### /home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/objects/base-objects

### /home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/hello-world

### /home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/frontend-java

### /home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/test

### /home/pi/Desktop/freedomotic-5.5.0.83-0504a4a/freedomotic-5.5.0.83-0504a4a/plugins/devices/restapi

### =========

### NetBeans Quick Tip #4 - Extending the Build Process

#### By Roman Strobl on [V 31, 2005](https://blogs.oracle.com/roumen/entry/netbeans_quick_tip_4_extending)

In today's tip I will show you how to extend the NetBeans build process on a simple example.   
  
Some background first - NetBeans utilizes Ant as it's primary project engine. So when you build any project the IDE executes Ant. Other Java IDEs use as primary project systems different engines, developed specially for these IDEs. That was the case with NetBeans 3.x as well but with NetBeans 4.0 it was changed to Ant which is de facto a standard for building Java applications.   
  
Ant is used similarly to GNU make to automate the build process. It can also handle build dependencies but unlike make it is completely plaftorm and shell independent (it's java-based), it's configuration is XML-based and can be extended easily. You can read more about Ant in it's [user manual](http://ant.apache.org/manual/index.html).   
  
Back to the tip - let's say I want to copy the jar file which is built with my project to a network share - for instance to publish it as a bleeding edge development build of the application. I want to create a backup copy of it's sources as well. Such tasks can be automated with NetBeans and Ant very easily.   
  
To do this I need to override the build.xml file which is stored in the main directory of my project. The build.xml file by default just includes a file called build-impl.xml, located in nbproject subdirectory. Build-impl should not be touched (it is generated automatically), instead the targets in build.xml need to be overriden.   
  
So I override the "-post-compile" target to create an archive with my sources using tar and gzip tasks. I want Ant to create a subdirectory called by current date and name the archive according to the date as well. To do that I at first initialize the DSTAMP variable using a task called <tstamp/>. I also override the "-post-jar" target to copy the jar which is created automatically in the dist directory:

<target name="-post-compile">

<tstamp/>

<tar tarfile="${dist.dir}/sources.tar" basedir="${src.dir}"/>

<gzip zipfile="${dist.dir}/sources.tar.gz" src="${dist.dir}/sources.tar"/>

<copy file="${dist.dir}/sources.tar.gz"

tofile="h:\\shared\\myapplication\\${DSTAMP}\\myapplication-sources-${DSTAMP}.tar.gz"/>

<delete file="${dist.dir}/sources.tar"/>

<delete file="${dist.dir}/sources.tar.gz"/>

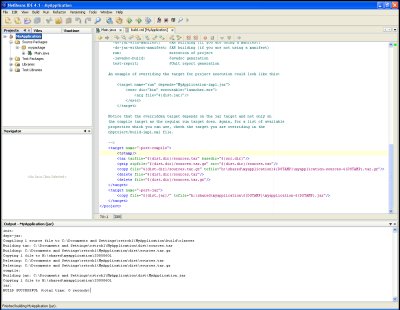
</target>

<target name="-post-jar">

<copy file="${dist.jar}/" tofile="h:\\shared\\myapplication\\${DSTAMP}\\myapplication-${DSTAMP}.jar"/>

</target>

Once I build the main project I get the following output:

[](http://blogs.sun.com/roller/resources/roumen/build_process_extend.png)

From now on everytime I build the project a copy of the jar and an archive with soures are created in a directory with current date on a network share. To achieve that I only needed to add few lines into the build.xml file (yeah, I know, you need to know which lines to add... ;-) If you didn't play with Ant yet, I suggest to try it, because it can help you automate a lot of tasks you may be doing manually. It's sometimes funny how much time we can give away to avoid doing a manual task - but with Ant this mostly pays off. Really.

### [samples-source-code](https://netbeans.org/projects/samples/sources/samples-source-code/show) / [samples](https://netbeans.org/projects/samples/sources/samples-source-code/show/samples) / [javaee](https://netbeans.org/projects/samples/sources/samples-source-code/show/samples/javaee) / [SimpleEE6App](https://netbeans.org/projects/samples/sources/samples-source-code/show/samples/javaee/SimpleEE6App) / [build.xml](https://netbeans.org/projects/samples/sources/samples-source-code/content/samples/javaee/SimpleEE6App/build.xml)

* [» File Revision History](https://netbeans.org/projects/samples/sources/samples-source-code/history?path%5B%5D=samples&path%5B%5D=javaee&path%5B%5D=SimpleEE6App&path%5B%5D=build.xml)
* [» Get Raw File](https://netbeans.org/projects/samples/sources/samples-source-code/content/samples/javaee/SimpleEE6App/build.xml?raw=true)

Top of Form

Bottom of Form

Size: 3434 bytes, 1 line

01.<?xml version="1.0" encoding="UTF-8"?>

02.<!-- You may freely edit this file. See commented blocks below for -->

03.<!-- some examples of how to customize the build. -->

04.<!-- (If you delete it and reopen the project it will be recreated.) -->

05.<!-- By default, only the Clean and Build commands use this build script. -->

06.<!-- Commands such as Run, Debug, and Test only use this build script if -->

07.<!-- the Compile on Save feature is turned off for the project. -->

08.<!-- You can turn off the Compile on Save (or Deploy on Save) setting -->

09.<!-- in the project's Project Properties dialog box.-->

10.<project name="SimpleEE6App" default="default" basedir=".">

11.<description>Builds, tests, and runs the project SimpleEE6App.</description>

12.<import file="nbproject/build-impl.xml"/>

13.<!--

14.

15.There exist several targets which are by default empty and which can be

16.used for execution of your tasks. These targets are usually executed

17.before and after some main targets. They are:

18.

19.-pre-init:                 called before initialization of project properties

20.-post-init:                called after initialization of project properties

21.-pre-compile:              called before javac compilation

22.-post-compile:             called after javac compilation

23.-pre-compile-single:       called before javac compilation of single file

24.-post-compile-single:      called after javac compilation of single file

25.-pre-compile-test:         called before javac compilation of JUnit tests

26.-post-compile-test:        called after javac compilation of JUnit tests

27.-pre-compile-test-single:  called before javac compilation of single JUnit test

28.-post-compile-test-single: called after javac compilation of single JUunit test

29.-pre-dist:                 called before archive building

30.-post-dist:                called after archive building

31.-post-clean:               called after cleaning build products

32.-pre-run-deploy:           called before deploying

33.-post-run-deploy:          called after deploying

34.

35.Example of pluging an obfuscator after the compilation could look like

36.

37.<target name="-post-compile">

38.<obfuscate>

39.<fileset dir="${build.classes.dir}"/>

40.</obfuscate>

41.</target>

42.

43.For list of available properties check the imported

44.nbproject/build-impl.xml file.

45.

46.

47.Other way how to customize the build is by overriding existing main targets.

48.The target of interest are:

49.

50.init-macrodef-javac:    defines macro for javac compilation

51.init-macrodef-junit:   defines macro for junit execution

52.init-macrodef-debug:    defines macro for class debugging

53.do-dist:                archive building

54.run:                    execution of project

55.javadoc-build:          javadoc generation

56.

57.Example of overriding the target for project execution could look like

58.

59.<target name="run" depends="<PROJNAME>-impl.jar">

60.<exec dir="bin" executable="launcher.exe">

61.<arg file="${dist.jar}"/>

62.</exec>

63.</target>

64.

65.Notice that overridden target depends on jar target and not only on

66.compile target as regular run target does. Again, for list of available

67.properties which you can use check the target you are overriding in

68.nbproject/build-impl.xml file.

69.

70.-->

71.</project>

Typically use a USB/Serial converter. There are two basic types. The first one is commonly used by people talking to microcontrollers and similar devices, it has a USB at one end, and a wiring connector (usually 4 or 6 pins or sockets) at the other, and is designed to use TTL level voltages.  
  
The second is a "true" USB/Serial converter and has USB at one end and a DB9 or DB25 (although not often these days!) connector at the other. The voltage levels at that end are RS-232 "standard" and should not be connected directly to anything on the Pi.  
  
Both kinds are freely available from on-line stores such as Amazon, here are UK links for the "true" serial kind...  
[http://www.amazon.co.uk/ABC-Products%C2 ... USB+serial](http://www.amazon.co.uk/ABC-Products%C2%AE-Converter-Convertor-Prolific/dp/B000Z3MBG8/ref=sr_1_1?ie=UTF8&qid=1370537129&sr=8-1&keywords=USB+serial)  
... and the TTL level kind...  
[http://www.amazon.co.uk/USB-Serial-TTL- ... +to+serial](http://www.amazon.co.uk/USB-Serial-TTL-Cable-FTDI/dp/B00514ZCHQ/ref=sr_1_10?s=electronics&ie=UTF8&qid=1370537179&sr=1-10&keywords=ftdi+usb+to+serial)  
  
When plugged in, the Pi will see the cable as a serial port, although I'm not entirely sure what name the device would have, except it won't be /dev/ttyAMA0 (/dev/ttyUSB0)  
Using lsusb or dmesg after plugging it in should be enlightening though.

If you issue an lsusb command, it will list the USB devices connected, and confirm that the device is recognised.   
Using dmesg will let you see the kernel messages, and will probably include the connection of the USB device, and should tell you the device name, which will be /dev/SOMETHING

For Example:

pi@RasPi1 /dev $ lsusb

Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.

Bus 001 Device 004: ID 0461:4d81 Primax Electronics, Ltd

Bus 001 Device 005: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge / myAVR mySmartUSB light

pi@RasPi1 /dev $ dmesg

[ 0.000000] Booting Linux on physical CPU 0

[ 0.000000] Initializing cgroup subsys cpu

… …

[ 1.869108] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002

[ 1.2246136] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1

[ 1.887786] usb usb1: Product: DWC OTG Controller

[ 1.896765] usb usb1: Manufacturer: Linux 3.6.11+ dwc\_otg\_hcd

[ 1.906796] usb usb1: SerialNumber: bcm2708\_usb

[ 1.916365] hub 1-0:1.0: USB hub found

[ 1.924410] hub 1-0:1.0: 1 port detected

… …

[ 2.662563] usb 1-1: New USB device found, idVendor=0424, idProduct=9514

[ 2.673959] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0

[ 2.686623] hub 1-1:1.0: USB hub found

[ 2.695099] hub 1-1:1.0: 5 ports detected

[ 2.982134] usb 1-1.1: new high-speed USB device number 3 using dwc\_otg

[ 3.092497] usb 1-1.1: New USB device found, idVendor=0424, idProduct=ec00

[ 3.107621] usb 1-1.1: New USB device strings: Mfr=0, Product=0, SerialNumber=0

[ 3.131966] smsc95xx v1.0.4

[ 3.206964] smsc95xx 1-1.1:1.0: eth0: register 'smsc95xx' at usb-bcm2708\_usb-1.1, smsc95xx USB 2.0 Ethernet, b8:27:eb:bf:36:7b

[ 3.322037] usb 1-1.2: new low-speed USB device number 4 using dwc\_otg

[ 3.449773] usb 1-1.2: New USB device found, idVendor=0461, idProduct=4d81

[ 3.464200] usb 1-1.2: New USB device strings: Mfr=0, Product=2, SerialNumber=0

[ 3.476438] usb 1-1.2: Product: USB Optical Mouse

[ 3.498232] input: USB Optical Mouse as /devices/platform/bcm2708\_usb/usb1/1-1/1-1.2/1-1.2:1.0/input/input0

[ 3.515258] hid-generic 0003:0461:4D81.0001: input,hidraw0: USB HID v1.11 Mouse [USB Optical Mouse] on usb-bcm2708\_usb-1.2/input0

[ 3.612067] usb 1-1.3: new full-speed USB device number 5 using dwc\_otg

[ 3.759066] usb 1-1.3: New USB device found, idVendor=10c4, idProduct=ea60

[ 3.781825] usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=3

[ 3.808577] usb 1-1.3: Product: CP2102 USB to UART Bridge Controller

[ 3.831822] usb 1-1.3: Manufacturer: Silicon Labs

[ 3.841395] usb 1-1.3: SerialNumber: 0001

[ 4.210162] udevd[153]: starting version 175

[ 5.508292] Registered led device: led0

[ 5.565822] usbcore: registered new interface driver usbserial

[ 5.692370] usbcore: registered new interface driver usbserial\_generic

[ 6.011989] USB Serial support registered for generic

[ 6.261058] usbserial: USB Serial Driver core

[ 6.479552] usbcore: registered new interface driver cp210x

[ 6.657298] USB Serial support registered for cp210x

[ 6.782337] cp210x 1-1.3:1.0: cp210x converter detected

[ 6.992161] usb 1-1.3: reset full-speed USB device number 5 using dwc\_otg

[ 7.185939] usb 1-1.3: cp210x converter now attached to ttyUSB0

[ 9.529531] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)

[ 9.972642] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)

[ 19.311214] smsc95xx 1-1.1:1.0: eth0: link up, 100Mbps, full-duplex, lpa 0xC1E1

A few years ago, a good friend of mine installed a small [photovoltaic system](http://en.wikipedia.org/wiki/Photovoltaic_system) on his roof. I'm very exited about installing some solar panels on a roof and start producing electricity. It turned out that the installed inverters have massive quality problems. So the idea was born to monitor the whole setup like i know it from computer systems. Each [inverter](http://en.wikipedia.org/wiki/Power_inverter) in this system has a serial port interface to transmit data.  
But at this time, the market of low cost and low power consuming computers wasn't distinct as it is nowadays. In February 2012, the Raspberry Pi Foundation started to sell Raspberry Pis. I saw this as a chance to get a cheap computer with a minimal power consumption.

So this blog post is about connecting a Raspberry Pi to inverters (or any other device) via RS232. We will use the [Java Communications API](http://www.oracle.com/technetwork/java/index-jsp-141752.html) to read received data from the inverters. We don't use the [integrated RS232 port](http://www.savagehomeautomation.com/projects/raspberry-pi-installing-a-rs232-serial-port.html) because we want to monitor more than just a single inverter, so we have to use USB-RS232 adapters.

## Prepare your Pi

To demonstrate this post, I'm using the following hardware setup:

* Raspberry Pi - Model B rev 2
* SDCard SanDisk Ultra SDHC Class 10
* [D-Link USB hub](http://www.dlink.com/us/en/home-solutions/connect/usb-hubs/dub-h7-7-port-usb-2-0-hub)
* Micro USB cable
* USB-RS232 adapters
* Alfa Networks USB WIFI adapter

## Connect Cables and Stuff

First of all we have to get a Pi ready running a Linux with SSH access. The Raspberry guys recommend using [Raspbian “wheezy”](http://www.raspberrypi.org/downloads) with Hard-Float. Follow the Quick start guide http://www.raspberrypi.org/quick-start-guide to setup your Pi.

Plug in all your USB devices into the USB hub, like keyboard, mouse, WIFI adapter, USB-RS232 adapters. The USB hub will be used as a power supply, so plug in the micro USB cable into the hub and into the micro USB port of your Pi. For the first boot, connect also a network cable into your Pi. Power up the USB hub and the Pi is going to boot for the first time.

## Configure WIFI

Now it's time to setup the WIFI connection. Install the wpasupplicant package via apt-get install wpasupplicant. Then copy paste /etc/network/interfaces and /etc/wpa\_supplicant/wpa\_supplicant.conf from my gists and adapt the files.

|  |  |
| --- | --- |
|  | #/etc/wpa\_supplicant/wpa\_supplicant.conf  ctrl\_interface=/var/run/wpa\_supplicant  eapol\_version=1  ap\_scan=1    network={  ssid="TheSSIDOfYourWlan"  scan\_ssid=1  proto=RSN  key\_mgmt=WPA-PSK  pairwise=CCMP  group=CCMP  psk="YourVeryStrongWPA2Key"  } |

[**view raw**](https://gist.github.com/grafjo/5116200/raw/3f97c024ca30d21fee7c392a7cde37e57e4ccb25/+wpa_supplicant.conf+) [**wpa\_supplicant.conf**](https://gist.github.com/grafjo/5116200#file-wpa_supplicant-conf) hosted with ❤ by [**GitHub**](https://github.com/)

|  |  |
| --- | --- |
|  | #/etc/network/interfaces  auto lo    iface lo inet loopback  iface eth0 inet dhcp    auto wlan0  iface wlan0 inet dhcp  wpa-driver wext  wpa-conf /etc/wpa\_supplicant/wpa\_supplicant.conf |

[**view raw**](https://gist.github.com/grafjo/5116200/raw/e0586539894117a8ef49405672984d28d8bf82a2/interfaces) [**interfaces**](https://gist.github.com/grafjo/5116200#file-interfaces) hosted with ❤ by [**GitHub**](https://github.com/)

## Serial Adapters

The D-Link USB hub has 7 ports (I define port 1 as the one on the left side and port 7 as the port on the right side) and I plugged two USB RS232 adapters into port 1 and port 2.

To check the USB-RS232 adapters are recognized by the Pi, run:

[?](http://blog.synyx.de/2013/03/running-rxtx-on-your-raspberry-pi/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | root@raspberrypi:/# lsusb | grep 232  Bus 001 Device 011: ID 0557:2008 ATEN International Co., Ltd UC-232A Serial Port [pl2303]  Bus 001 Device 010: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC  root@raspberrypi:/# ls /dev/ttyUSB\*  /dev/ttyUSB0  /dev/ttyUSB1  root@raspberrypi:/# |

One important question is, which of the adapters is /dev/ttyUSB0? Smart people will say, /dev/ttyUSB0 is the one, plugged in first - that's right.  
Lazy people (including myself) are using udev rules to create an alias per USB hub port.

|  |  |
| --- | --- |
|  | #/etc/udev/rules.d/70-persistent-usb-rs232.rules  KERNEL=="ttyUSB\*", KERNELS=="1-1.3.1.1", SYMLINK="ttySerialPort1"  KERNEL=="ttyUSB\*", KERNELS=="1-1.3.2", SYMLINK="ttySerialPort2"  KERNEL=="ttyUSB\*", KERNELS=="1-1.3.1.2", SYMLINK="ttySerialPort3"  KERNEL=="ttyUSB\*", KERNELS=="1-1.3.3", SYMLINK="ttySerialPort4"  KERNEL=="ttyUSB\*", KERNELS=="1-1.3.1.3", SYMLINK="ttySerialPort5"  KERNEL=="ttyUSB\*", KERNELS=="1-1.3.4", SYMLINK="ttySerialPort6"  KERNEL=="ttyUSB\*", KERNELS=="1-1.3.1.4", SYMLINK="ttySerialPort7" |

After the rules are applied, all serial ports are accessible via aliases and we have a nice and easy way accessing our serial ports, no matter which port was plugged in first.

|  |  |
| --- | --- |
| 1  2  3 | root@raspberrypi:/# ls -l /dev/ttySerialPort\*  lrwxrwxrwx 1 root root 7 Jan  1  1970 /dev/ttySerialPort1 -> ttyUSB1  lrwxrwxrwx 1 root root 7 Jan  1  1970 /dev/ttySerialPort2 -> ttyUSB0 |

To run the Java application later without root, we need a new user which has the privilege to access serial ports. To do so, just add a new user to group dialout

|  |  |
| --- | --- |
| 1 | root@raspberrypi:/# useradd -m -d /home/rxtxpi -s /bin/bash -G dialout rxtxpi |

## Installing Software

Now we're going to install the required software to communicate via serial adapters. For Java applications, we can use the Java Communications API (JCA). JCA requires some native platform specific code which is called via Java Native Interface (JNI). We also need the Java part of JCA.

The ugly part - JCA is not provided by Oracle via JRE and I couldn't find any downloads from Oracle. The good part - there is an Open Source implementation of JCA called [RXTX](http://rxtx.qbang.org/). The native part can be installed via librxtx-java deb package. As long as there is no RXTX jar in Maven Central you can use this <https://github.com/grafjo/rxtx> as the Maven repository for RXTX.

We just install a JRE and RXTX on Pi, because compiling software on the Pi is really really slow! Install a Java 1.6 JDK and Gradle > 1.2 on your desktop, build the Java application there and copy it to the Pi.

|  |  |
| --- | --- |
| 1 | root@raspberrypi:/# apt-get install openjdk-6-jre librxtx-java |

Now the Pi is ready to run a Java application :-)

## Running Sample Application

The sample application just displays the plugged in serial adapters of your Pi. So switch to your desktop and build the sample application:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | joo@jgraf:~$ git clone git@github.com:grafjo/rxtxpi.git  Cloning into 'rxtxpi'...  remote: Counting objects: 25, done.  remote: Compressing objects: 100% (16/16), done.  remote: Total 25 (delta 1), reused 24 (delta 0)  Receiving objects: 100% (25/25), done.  Resolving deltas: 100% (1/1), done.  joo@jgraf:~$ cd rxtxpi/  joo@jgraf:~/rxtxpi$ gradle distZip  :compileJava  warning: [options] bootstrap class path not set in conjunction with -source 1.6  1 warning  :processResources UP-TO-DATE  :classes  :jar  :startScripts  :distZip  BUILD SUCCESSFUL  Total time: 4.458 secs |

Copy the application to your Pi:

|  |  |
| --- | --- |
| 1 | joo@jgraf:~/rxtxpi$ scp build/distributions/rxtxpi.zip rxtxpi@raspberry:/home/rxtxpi |

Switch back to your Pi, unzip the rxtxpi.zip. To use the predefined aliases for serial ports, we have to modify the default JVM options to DEFAULT\_JVM\_OPTS="-Dgnu.io.rxtx.SerialPorts=/dev/ttySerialPort1:/dev/ttySerialPort2" inside the rxtx\_pi startup script rxtx\_pi/bin/rxtx\_pi.

Ok - when every thing was installed successfully, we will get this output from our Java application using RXTX

|  |  |
| --- | --- |
| 1  2  3  4  5 | rxtxpi@raspberrypi ~/rxtx\_pi/bin $ ./rxtx\_pi  [main] INFO de.synyx.rxtxpi.SerialPortUtils - Looking for serial ports  [main] INFO de.synyx.rxtxpi.SerialPortUtils - Found port: /dev/ttySerialPort1  [main] INFO de.synyx.rxtxpi.SerialPortUtils - Found port: /dev/ttySerialPort2  rxtxpi@raspberrypi ~/rxtx\_pi/bin $ |

So this blog post was about getting a Pi ready to run a RXTX Java application and verified that everything works well. The next post will be about how to use the Java Communications API to read data from an inverter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | It's possible you can create a script java\_server\_launch.sh like this:  #! /usr/bin/sh  PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin  JAVA=/usr/bin/java  MY\_SERVER=/home/your\_username/DocumentManager/DocumentServer-0.2.jar  USER=your\_username  /bin/su - $USER -c "$JAVA -jar $MY\_SERVER &"  Put your script under /etc/init.d directory, and then use the command:  update-rc.d java\_server\_launch.sh defaults  more on update-rc.d command by using man update-rc.d.  Hope this help.   |  |  |  | | --- | --- | --- | |  |  |  | |
|  | |  |  | | --- | --- | |  | Hi, I tried that and its not working. I ssh'd in and looked at the processes running and its not there. Do i need to put in a start script somewhere? – [Ben Flowers](http://stackoverflow.com/users/1287962/ben-flowers) [Aug 4 '12 at 14:26](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694107_11809315) | |  | @BenFlowers : hello, yes under /etc/init.d directory – [TOC](http://stackoverflow.com/users/1560320/toc) [Aug 4 '12 at 14:27](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694122_11809315) | |  | Okay so basically i created the java\_server\_launch.sh file and then ran the commands you said. Is there anything else i need to do? (still not working) thanks for your help btw – [Ben Flowers](http://stackoverflow.com/users/1287962/ben-flowers) [Aug 4 '12 at 14:34](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694204_11809315) | |  | @BenFlowers : I edited my post, by the way you can take a look at /etc/init.d/skeleton file to see how you can use it to meet your need. – [TOC](http://stackoverflow.com/users/1560320/toc) [Aug 4 '12 at 14:51](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694376_11809315) | |  | still not working, its close i can tell!! so my jar file is located at ~/DocumentManager/DocumentServer-0.2.jar and i am sshing in as ubuntu@(amazoninstance). im storing my script.sh file in /etc/init.d and it looks like this: #! /usr/bin/sh PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin JAVA=/usr/bin/java MY\_SERVER=~/DocumentManager/DocumentServer-0.2.jar USER=ubuntu /bin/su - $USER -c "$JAVA -jar $MY\_SERVER &" and running update-rc.d script.sh defaults any clue what im doing wrong? – [Ben Flowers](http://stackoverflow.com/users/1287962/ben-flowers) [Aug 4 '12 at 15:18](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694667_11809315) | |  | FIXED IT!! just needed to chmod +x the script file D'OH! – [Ben Flowers](http://stackoverflow.com/users/1287962/ben-flowers) [Aug 4 '12 at 15:28](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694799_11809315) | |  | @BenFlowers : Ok, i was thinking you've done this before :). – [TOC](http://stackoverflow.com/users/1560320/toc) [Aug 4 '12 at 15:33](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694857_11809315) | |  | never underestimate the stupidity of someone who has been trying to fix a small problem for 3 hours! Thanks again! – [Ben Flowers](http://stackoverflow.com/users/1287962/ben-flowers) [Aug 4 '12 at 15:33](http://stackoverflow.com/questions/11809191/linux-launch-java-program-on-startup-ec2-instance#comment15694865_11809315) | |  |  | |