

DukePad



The DukePad is a Do-It-Yourself, make-at-home tablet computer based on the [Raspberry Pi](#) and JavaSE Embedded 8. The plans and instructions for building the DukePad are available [here](#), and we're working with suppliers to make available pre-made kits that can be more easily assembled. The software on the DukePad uses [Raspbian](#) Linux as the operating system, and an OSGi-based JavaFX environment. Within this DukePad environment, apps are simple JavaFX OSGi Modules.

The DukePad is not a product, it is an open source, freely available set of plans and software for assembling your own tablet using off the shelf components. As such, the quality of the DukePad software environment is demo-quality (although we did strive to write as much real functionality as we could, the realities of demo presentations requires sacrificing time on parts of the applications that are not going to be shown, in favor of smoothing out those parts that will be shown). The code is hosted in the [OpenJFX repositories](#) under apps/experiments/DukePad. We hope to see forks of this code (GitHub, BitBucket, whatever you like best) and lots of experimentation and improvement that can be shared.

This guide assumes that your Raspberry Pi has "dukepad" as the host name, although of course you can use anything (including the raw IP address). The real work starts with step 3, building the DukePad software. One thing to keep in mind, is that although the Raspberry Pi can run X11, we will not normally use it, since JavaFX is going to take over the entire screen. When downloading software to the PI, you can choose either to start up X (startx), or you can choose to download onto your desktop system and sftp / scp the files over the PI. This guide assumes you know how to get files onto the PI (and if not, a quick lesson on sftp / scp should do the trick).

Software

First, setup your new Raspberry PI as usual. Here are [quick start instructions](#) if needed. The PI needs to be configured so that a [sufficient amount of memory is devoted to VRAM](#) (graphics memory, also called Video Core). We used an even split of 256MB for VRAM and 256MB for system memory. You can tweak the amount of VRAM used by FX (see the command line args sent to DukePad below), so if you only have 256MB total you might be able to make it work with a 128MB/128MB split, with some tuning, although we have not attempted it.

If your installation does not already have it, download and install the latest JavaSE Embedded release. As of Sep. 2013, JavaSE Embedded is only available as a [weekly build](#). After Mar 2013, you should [just use the official builds](#). You are looking for Linux ARMv6/7 VFP, HardFP ABI. This is the version of Java that will work on a PI.

Documentation

- [JDK Docs \(61.80 MB zip | HTML\)](#)
- [JavaFX Docs \(5.29 MB zip | HTML\)](#)

	Platforms	JRE	JDK
Windows	32-bit	exe (md5) 29.34 MB	exe (md5) 104.24 MB
	64-bit	exe (md5) 32.74 MB	exe (md5) 108.29 MB
Mac OS X	64-bit	dmg(md5) 59.78 MB	dmg (md5) 164.77 MB
Linux	32-bit	tar.gz (md5) 55.11 MB	tar.gz (md5) 105.40 MB
	64-bit	tar.gz (md5) 54.11 MB	tar.gz (md5) 104.38 MB
Linux ARMv6/7 VFP, HardFP ABI	32-bit		tar.gz(md5) 82.94 MB
Solaris SPARC	32-bit	tar.gz (md5) 61.06 MB	tar.gz (md5) 105.98 MB
	64-bit*	tar.gz (md5) 18.63 MB	tar.gz (md5) 18.81 MB
Solaris	32-bit*	tar.gz (md5) 102.02 MB	tar.gz (md5) 102.02 MB
	64-bit*	tar.gz (md5) 16.23 MB	tar.gz (md5) 16.40 MB

Installation simply involves uncompressing the downloaded file and putting it in a directory of your choosing. For example:

Installing JavaSE Embedded

```
pi@dukepad$ sudo su
root@dukepad:~# tar -xzf jdk-8-ea-b108-linux-arm-vfp-hflt-18_sep_2013.tar.gz
root@dukepad:~# mv jdk1.8.0 /opt/
```

At this point, the JDK has been installed into `/opt/jdk1.8.0` and includes JavaFX as well. Next, we need to install a few additional Raspberry PI packages in order to play media, and to configure the splash screen and auto-booting. You can skip the splash screen / auto-boot instructions if you are just playing with the DukePad software and not creating a table device.

First, get the proper packages:

apt-get

```
root@dukepad:~# apt-get update
root@dukepad:~# apt-get install fbi
root@dukepad:~# apt-get install mpg321
```

The "fbi" package is required for the boot loading screen, and the "mpg321" package is for being able to play media.

Now we will walk through the steps for downloading and building the DukePad JavaFX software and apps. The sources are held in the OpenJFX project repositories. You need to clone OpenJFX, but you don't need to build all of OpenJFX, but can just build the DukePad independently. On your desktop, clone <http://hg.openjdk.java.net/openjfx/8/graphics/rt>. If you are using IntelliJ IDEA, locate under `rt/apps/experimental` the DukePad project. Open it and build. It should build, auto-downloading dependencies from maven central. Alternatively, you can try using the gradle build file. To do so, `cd` into `rt/apps/experimental/DukePad`. Execute:

apt-get

```
me@my-computer:~$ gradle artifacts
```

(For a while the IDEA project build wasn't working, so the gradle script was thrown together. The gradle script isn't being used by anybody, so it may have issues. Ultimately, gradle will download the dependencies into the lib/ directory. You can manually populate it if necessary).

After a successful build, the following libraries should be in rt/apps/experimental/DukePad/lib:

apt-get

```
grizzly-framework-2.3.1.jar
grizzly-http-2.3.1.jar
grizzly-http-server-2.3.1.jar
grizzly-rcm-2.3.1.jar
tyrus-core-1.0.jar
tyrus-spi-1.1.jar
tyrus-client-1.0.jar
tyrus-core-1.1.jar
tyrus-websocket-core-1.0.jar
tyrus-client-1.1.jar
tyrus-server-1.0.jar
tyrus-websocket-core-1.1.jar
tyrus-container-grizzly-1.0.jar
tyrus-spi-1.0.jar
pi4j-native-0.0.5-soft-float.so
pi4j-core-0.0.5.jar
pi4j-native-0.0.5-hard-float.so
javax.json-1.0.1.jar
javax.websocket-api-1.0.jar
```

The actual built artifacts are in DukePad/build/artifacts. Everything in this directory must be uploaded to the Raspberry PI. The following example will use SCP to copy the files from build/artifacts up to a directory on the pi called /opt/dukepad (you may have had to pre-create this directory on the server and assign full rwx permissions on it).

apt-get

```
me@my-computer:~$ scp -r build/artifacts/* pi@dukepad:/opt/dukepad
```

Finally, back on the PI we need to create a shell script which will start up the DukePad UI. We're using this as our script:

apt-get

```
root@dukepad4:/opt/dukepad# vi run.sh

echo "Using $JAVA_HOME"
cd /opt/dukepad
$JAVA_HOME/bin/java \
    -Dcom.sun.javafx.transparentFramebuffer=true \
    -Dprism.verbose=true \
    -Degl.depthSize=24 -Dprism.glDepthSize=24 \
    -Dcom.sun.javafx.experimental.embedded.3d=true \
    -Di2c.bus=0 -Di2c.address=0x68 \
    -Dprism.maxvram=209717200 \
    -Dprism.targetvram=100663296 \
    -Xmx90m \
    -Xms90m \
    -DuseMag=false \
    -jar dukepad.jar
```

We played with a lot of different command line values (for example, for remote debugging, JMX monitoring, etc). The maxvram and targetvram values are in bytes, and indicate how much of VRAM JavaFX will use for graphics. We didn't fine tune these values, just picked something that seemed reasonable, so there might be the ability to tweak these values for better performance (although it doesn't seem to be highly likely). We set the VM heap size to 90m so as to make startup a bit better, so that instead of the VM running out of space during startup it would just start off with a big chunk of memory. The OS still has another 150MB or so to play with, which is probably excessive, but the DukePad FX app didn't require a lot of heap (a lot less than 90MB actually), so we didn't see a reason to make it really large.

If you execute this script at this point, it should start up the DukePad, with whatever applications you have in your /opt/dukepad/apps directory. A quick note, if you have any MP3 files for music or MP4 files for media or JPG files for images, then put them in /opt/dukepad/media/images, /opt/dukepad/media/movies, /opt/dukepad/media/music as appropriate. If you want preview thumbnails for movies, also include in the movies directory medianame-thumb.jpg, where medianame is the name of the media file. For example, IronMan.mp4 would have IronMan-thumb.jpg.

If you want to enable auto-start, so that when the DukePad is booted up it will automatically launch the UI, then do the following:

apt-get

```
root@dukepad:~# nano /etc/inittab
```

Scroll down and edit the file as thus:

apt-get

```
#1:2345:respawn:/sbin/getty 115200 tty1
1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1
```

Edit this file in /etc/init.d

apt-get

```
root@dukepad:~# vi /etc/init.d/asplashscreen

#!/bin/sh
### BEGIN INIT INFO
# Provides:          asplashscreen
# Required-Start:
# Required-Stop:
# Should-Start:
# Default-Start:    S
# Default-Stop:
# Short-Description: Show custom splashscreen
# Description:      Show custom splashscreen
### END INIT INFO

do_start () {
    echo "Starting Duke Pad"
    /usr/bin/fbi -T 1 -noverbose -a /opt/dukepad/Duke-Startup.jpg
    exit 0
}

case "$1" in
    start|"")
        do_start
        ;;
    restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
    stop)
        # No-op
        ;;
    status)
        exit 0
        ;;
    *)
        echo "Usage: asplashscreen [start|stop]" >&2
        exit 3
        ;;
esac

:
```

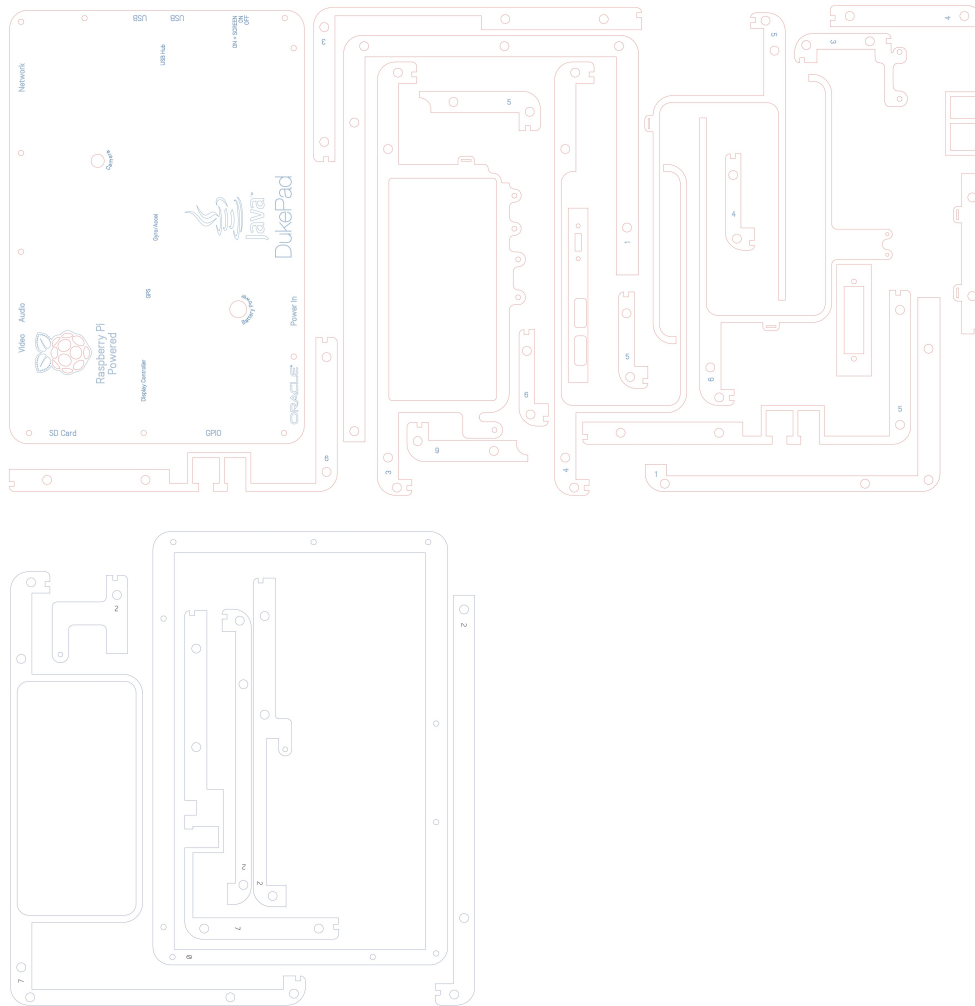
And finally, edit /etc/rc.local before "exit"

apt-get

```
# START DUKE PAD
nohup /opt/dukepad/run.sh > /opt/dukepad/log.txt &
```

Hardware

OK, now for the hardware for building the DukePad. Some of the apps require hardware setup. There are some other steps not yet documented on this wiki for setting up accelerometer support. The following two files provide all of the template for laser cutting acrylic for the body of the DukePad. There are two different thicknesses, 3mm(1/8") and 4.5mm (3/16"). We're working with [Special Computing](#) to provide pre-built kits that can be ordered. This wiki will be updated with further information as it becomes available.



- Screen + Display Board \$135 http://www.chalk-elec.com/?page_id=1280#!/~/product/category=3094861&id=14647624
- Raspberry Pi Model B \$35 - <http://www.newark.com/raspberry-pi/raspbrry-modb-512m/model-b-assembled-board-only/dp/43W5302>
- Raspberry Pi Camera \$25 - <http://www.newark.com/jsp/search/productdetail.jsp?sku=69W0689>
- 16Gb Mem Card \$13 - http://www.amazon.com/gp/product/B003VTKNEQ/ref=oh_details_o01_s00_i00?ie=UTF8&psc=1
- USB Wifi \$10 - http://www.amazon.com/gp/product/B003MTTJOY/ref=oh_details_o03_s00_i00?ie=UTF8&psc=1
- USB Hub \$20 - http://www.amazon.com/gp/product/B005P2BY5I/ref=oh_details_o09_s00_i01?ie=UTF8&psc=1
- Battery Pack \$40 - http://www.amazon.com/gp/product/B009USAJCC/ref=oh_details_o03_s00_i01?ie=UTF8&psc=1
- MPU 9150 Gyro/Accel/Mag \$50 - <https://www.sparkfun.com/products/11486>
- 1ft USB A to MiniUSB B \$4 - http://www.amazon.com/gp/product/B002L5U7N2/ref=oh_details_o04_s00_i01?ie=UTF8&psc=1
- 20x #4-40 7/16" Flat head screw \$1 - http://www.amazon.com/gp/product/B000FN7UQO/ref=oh_details_o07_s01_i01?ie=UTF8&psc=1
- 10x Standoffs Hex Female To Female 3/16" 4-40 3/4" Aluminum \$3.5 - <https://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?catalogId=10001&langId=-1&productId=133656&storeId=10001&krypto=vFgRhKXJZKuM8slfuPRmxv3tw%2BNp1UpM3AZ3S0xdP278VMhIjwnqg%3D%3D&ddkey=https:StoreCatalogDrillDownView>
- Panel Mount RJ45 \$5 - <http://www.adafruit.com/products/909>
- USB A connector for power from battery \$1 - <http://www.adafruit.com/products/1387>
- DB25 IDC Female for external GPIO \$2.5 - http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?search_type=jamecoall&catalogId=10001&freeText=115976&langId=-1&productId=115976&storeId=10001&ddkey=http:StoreCatalogDrillDownView
- GPIO Cable, cut one end of and replace with DB25 \$3 - <http://www.adafruit.com/products/862>
- Case Acrylic \$15
- 6x #4-40 1/2 Flat Bolt and Nut \$1
- 4x #2-56 5/16" Bolt and Nut \$1
- Small amount of wire for power wiring and i2C \$1
- 650uf 6.3v capacitor to keep Pi running while switching screen on/off \$0.3
- 2P3R Slide Switch \$1.25

Credits

We'd like to give a special thanks to the following individuals / companies for helping us out in this project, either directly or indirectly. First thanks to Eben Upton and the Raspberry Pi foundation, for giving us the Pi in the first place, and for taking time out to talk over with us technical details that helped us make JavaFX work well on the Pi. We'd like to thank Bill at Special Computing for working with us to make available dev kits of the DukePad for customers.

Also, thanks to the [PiBow](#) for an amazing enclosure for normal raspberry Pis, and the inspiration for the design of the DukePad case. And thanks to Rich, Jasper, Alexander K, Stas, Debbie, Gary, John Yoon, and all the others who have helped in producing the hardware and software for the DukePad (and our management for supporting us in this wild JavaOne keynote idea!). We hope it will be useful to the Raspberry PI community, particularly in the Educational markets as a way for kids to learn how computers work and to fire their imaginations for those things that are not yet built, but desperately need a builder to dream them up and make them happen!