The Mathematical Function Atan2 • Resizing your Android partitions



Magazine

DREAM MACHINE AND WHISPER

AN INSIDE LOOK AT PREBUILT UBUNTU 13.04 AND 13.10 DISK IMAGES THAT GET YOUR ODROID UP AND RUNNING FAST!



Ubuntu Mathematics Tools

Lubuntu Whisper 13.10 Saucy Salamander

Blackbox

LXDE

XBMC PICTUR

PLUS:

- PACKAGE YOUR OWN CUSTOM COMPILED SOFTWARE
- PROGRAM YOUR ODROID SHOW WITH REBOL
- ENABLE MULTI-CHANNEL AUDIO OUTPUT WITH XBMC



What we stand for,

We strive to symbolize the edge technology, future, youth, humanity, and engineering.

Our philosophy is based on Developers. And our efforts to keep close relationships with developers around the world.

For that, you can always count on having the quality and sophistication that is the hallmark of our products.

Simple, modern and distinctive. So you can have the best to accomplish everything you can dream of.





We are now shipping the ODROID U3 devices to EU countries! Come and visit our online store to shop!

Address: Max-Pollin-Straße 1 85104 Pförring Germany

Telephone & Fax phone : +49 (0) 8403 / 920-920 email : service@pollin.de

Our ODROID products can be found at: http://www.pollin.de/shop/suchergebnis.html?S_ TEXT=odroid&log=internal



EDITORIAL

ver since we began publishing ODROID Magazine, we've been receiving some great articles from our international contributors involving their innovative hardware projects, including the Giant Android Tablet, ODROID Truck PC, and the Unmanned Ground Vehicle. The worldwide maker community has shown that

there is almost no limit as to what an ODROID can do.

Because many ODROIDians are involved in software projects as well, we plan to feature some of the many excellent OS images and software that the ODROID community and Hardkernel developer team have been producing in an ongoing series called OS Spotlight.

We kicked off the column last month with an overview

of the software included with the Fully Loaded Ubuntu 12.11 image, and this month we are highlighting Dream Machine, which is ideal for office work, programming, media playback, and anything else where a traditional desktop PC would be used.

As of June 1st, there are now over 5000 members in the ODROID forums at http://forum.odroid.com, showing a phenomenal growth of 500% over the last 1.5 years. With so much interest in the ODROID platform, the original version of U3 actually sold out last month, most likely due to its small footprint and very affordable price. To respond to the increased demand for the compact model, Hardkernel will soon begin shipping its newest generation of the U series hardware called the ODROID-U3+, containing several new features: http://bit.ly/1p8zi6L.

For Android enthusiasts, Hardkernel published a sneak peek Beta version of Android KitKat 4.4, which you can download and test at http://bit.ly/lhsFdRB. It's still a work in progress, so make sure to post any bugs and/or feedback on the ODROID forums.

In response to many requests for HDMI Passthrough DTS audio, there is a new inexpensive piece of hardware, now available from the Hardkernel Store, called the USB-S/PDIF. The USB-S/PDIF sends DTS and AC3 digital encoding via the HDMI cable to your audio receiver, delivering crisp 5.1 surround sound to your living room or office via your ODROID running Linux and XBMC 13. It's now available for \$16 at http://bit.ly/lkIt8aC.

Hardkernel has not only been upgrading its Ubuntu images to 14.04, but has also improved the XBMC experience with their long-awaited official 1080p release of the popular XBMC-based Media Center image. The new version of XBMC for Linux, available on all platforms, takes advantage of the recent improvements to video decoding in the 3.8 kernel and delivers smooth 720p and 1080p video for all of the encodings supported by the Mali GPU and VPU hardware. Keep in mind that it is an experimental build, but has been reported to work very well. It's a great distribution for those who previously used OpenElec or Xbox as a set-top box. Check out the build at http://bit.ly/11fkYpN.

ODROID Magazine, published monthly at http://magazine.odroid.com, is your source for all things ODROIDian. Hard Kernel, Ltd. • 704 Anyang K-Center, Gwanyang, Dongan, Anyang, Gyeonggi, South Korea, 43I-8I5 Makers of the ODROID family of quad-core development boards and the world's first ARM big.LITTLE architecture based single board computer.



Join the ODROID community with members from over I35 countries, at http://forum.odroid.com, and explore the new technologies offered by Hardkernel at http://www.hardkernel.com.

ODROID Magazine

Rob Roy, Chief Editor

I'm a computer programmer living and working in San Francisco, CA, design-

ing and building web applications for local clients on my network cluster of ODROIDs. My primary languages are jQuery, Angular JS and HTML5/CSS3. I also develop pre-built operating systems, custom kernels and optimized applications for the ODROID platform based on Hardkernel's official releases, for which I have won several Monthly Forum Awards. I use my ODROIDs for a variety of purposes, including media center, web server, application development, workstation, and gaming console. You can check out my 100GB collection of ODROID software, prebuilt kernels and OS images at http://bit.ly/lfsaXQs.



Lechnowsky, Editor

I am President of Respectech, Inc., а technology consultancy in Ukiah, CA, USA that I founded in 2001. From my background in electronics and computer programming, I manage a team of technologists, plus develop custom solutions for companies ranging from small businesses to worldwide corporations. ODROIDs are one of the weapons in my arsenal for tackling these projects. My favorite development languages are Rebol and Red, both of which run fabulously on ARM-based systems like the ODROID-U3. Regarding hobbies, if you need some, I'd be happy to give you some of mine as I have too many. That would help me to have more time to spend with my wonderful wife of 23 years and my four beautiful chil-

dren. Br cho to

Bruno Doiche, Art Editor

Secured his computing necro- mancy skills after bringing a fiber optics switch back to life, getting his Macintosh back from death, getting his fiancee's T400 back from death (which was a old-style dd data transfer), and managing how to handle the cold innards of his steady job data center.

News from Art Editor Bruno:

Getting back to our regular schedule is no small task. After a couple of minor setbacks, we throttled hard to get to a speedier rhythm, but not without having, as usual, the best fun! What were our trade offs? Well... we didn't manage to pack our astounding amount of tips, but now we nailed the code color palette and type size!. It's easy to read and good looking!

The cover was also a challenging idea, as when we were chatting and defined that the centerpiece cover article would be Rob's Linux distros. I had this micro obsession on getting all current production ODROIDS together, but believe or not, I don't have the actual production triad (U3, X2, XU). But as easy as seems to get them side by side in a single photo, there are plenty of ODROIDians that never seen them together. But no longer! I have a trusty friend that has three of them, and sent me a photo of three of his ODROIDS for me to put on the cover. So here is my thanks Patola, and I hope to feature you in an interview regarding 3D printing using ODROIDS!

What else is new? Now, Rob is working directly on the art master files! We have gotten so much more agile on the revision process, and are as ever, working hard to deliver to you, our trusty readers a hard-fact, spot-on magazine that we love to do! Not that we don't get some crazy issues now and then!





PROGRAMMING YOUR ODROID-SHOW

USING THE REBOL PROGRAMMING LANGUAGE TO IMPROVE THE HARDWARE INTERFACE

By Bo Lechnowsky

he ODROID-Show module is incredibly useful for many tasks. For instance, it is a great device to display information that can be viewed at a glance. However, it can be cumbersome to display the information you want in an easy manner.

To make things simpler, I used the incredible capabilities of Rebol to create an easy dialect to allow displaying many types of information on the ODROID-Show. Because Rebol is cross-platform, the same scripts will run on any operating system.

You can also connect multiple ODROID-Show modules to a single ODROID, or any other computer with USB ports. This dialect will allow easy handling of that.

The function to initiate the dialect is called oshow (The word show is already used in Rebol to refresh GUI displays).

I've highlighted the code that is sent to the ODROID-Show module via the oshow dialect with bold letters. As is well known, bash scripting isn't the easiest to read, especially when interspersed with control codes. In addition, the Rebol example above is about 35% smaller, allows any order of color cycling, allows any number of words of any length at any number of coordinates, and will run on any platform that has a Rebol 3 interpreter (pretty much everything except for Windows RT and IOS which will come soon). I leave it to the reader to decide which is easier to modify.



Using the oshow dialect, compare the program written in Rebol to the example provided in bash scripting in Tutorial #1: Text Output from the ODROID-Show user guide:

do %oshow.r3
call {port_open}
<pre>serialport: %/dev/ttyUSB0</pre>
display-text: ["ODROID" 25x100 "SHOW" 55x150]
oshow serialport [<i>reset rotate 0</i>]
forever [
foreach color [red green yellow blue magenta cyan white] [
foreach [word position] display-text [
oshow serialport compose [cursor to (position) font (color) 5]
foreach letter word [
oshow serialport compose [(to-string letter)]
wait .02
1
]
1
]

TECHNICAL ARTICLE

Here is the full dialect definition.

font optionally followed by: -

black	
red	optionally followed by:
green	default (sets default color
yellow	as foreground color)
blue	
magenta	optionally followed by:
cyan	1 (sets text size to: 6x8 pix-
white	els * the number specified here)

Examples:

white

oshow [font cyan] oshow [font white default] oshow [font 2] (sets text size to 12x16 pixels) oshow [font 1 cyan "Hello " 2 red "ODROID!"]

background followed by one of the colors listed under foreground coloring.

optionally followed by: default (sets default color as background color)

Examples:

oshow [background black] oshow [background yellow default]

cursor followed by:

down
up
position? (reports current cursor position)
save (saves the current cursor position)
restore (restores the cursor to the position previously saved)
to (followed by:)
3x5 (moves the cursor to column 3, row 5)
home (moves cursor to column 1, row 1)
bottom (moves cursor to row 1 of the same column)
reset (resets the LCD screen)
clear (clears the screen)
off (turns LCD screen off)
on (turns LCD screen on)
key (simulated keyboard press) followed by:
2 (optional: the number of keypresses - default is 1)
up
down
left
right
rotate (rotates the display in 90 degree increments) followed by:
left (counterclockwise 90 degrees)
right (clockwise 90 degrees)
draw (not yet functional) followed by:
10x20 (optional: the upper left corner of the image - default is $0x0$
50x50 (optional: the size of the image - default is 320x240)
<image data=""/>

RECOMPILING MALI DRIVERS **UPDATING TO THE** LATEST RELEASE (R4P0-00RELI)

by Mauro Ribeiro

repare your ODROID to run the latest R4P0 Mali drivers by typing the following into a Terminal window:

sudo apt-get install xserverxorg-video-armsoc xserver-xorgdev libudev-dev

As of May 5, 2014, Hardkernel is no longer supporting the Mali r3p2 drivers, and haved moved on to using the R4P0 version exclusively. There are two major changes:

1. Instead of using the Mali DDX, the Armsoc driver is installed.

2. UMP no longer exists, and Mali is now using dma-buf.

We have provided an updated Armsoc DDX in order to adapt your existing Ubuntu or Debian operating system to the new Mali drivers. The source code for the r4p0 branch of the repository is located at http://bit.ly/loySeYE. Download the source to a temporary folder, then type make install at the command prompt after navigating to the temporary folder.

Blobs for EGL/GLESv1/GLESV2 are available for free download at http:// bit.ly/1hiwJw9, which includes an optional example xorg.conf file.

Note that if your defconfig still has UMP, it will probably fail to build, so make sure to remove it first.

If you have questions regarding the new Mali drivers, feel free to ask them in the ODROID community forums at http://forum.odroid.com/.

TIPS AND TRICKS

GOT WIIMOTE? MAKE YOURSELF AN AWESOME GYROSCOPIC MOUSE

by Rob Roy

o you have a Wii remote that hasn't seen much use lately? It's actually a very elegant piece of input hardware, with an ergonomic design, infrared scanner and gyroscope sensors. Ubuntu has several packages available from its repositories that automatically read the sensors in the Wii and translate them into mouse and button movements. To install the driver software, type:

\$ sudo apt-get install libcwiid1 lswm wmgui wminput

After installation has completed, connect to the Wii remote by pressing 1 and 2 simultaneously, which activates the Bluetooth Connect mode. Once the Wiimote's lights start flashing, type:

\$ hcitool scan

or

\$ lswm

Either command will display the MAC address for all nearby Wiimote controllers that are currently in Bluetooth Connect mode. Load the Wiimote driver by typing the following, using the MAC address from the previous step. In this example, the MAC address is 01:FA:2C:9D:BB:05:

\$ sudo wminput \
01:FA:2C:9D:BB:05

The wiimote will then be treated by Ubuntu as a gyroscopic air mouse, suitable for use in situations where a normal mouse won't work, such as while delivering a group presentation or sitting on a couch.

TECHNICAL ARTICLE

And here is the code for the oshow dialect, to be saved as "oshow.r3", or it can be downloaded from the URL link in the header below.

```
REBOL [
       Title: {ODROID Show Dialect}
       Date: 20-May-2014
       Author: {Bohdan Lechnowsky}
       URL: http://respectech.com/odroid/odroid-show.r3
       Purpose: {
          Provides a dialect to allow easy usage of the ODROID Show hard-
ware component
       3
   1
  oshow: func [
       {Allow easy utilization of the ODROID-Show display module}
         serport {Name of the serial port to send to (i.e. "serial://
ttyUSB0/57600" or "%/dev/ttyUSB0")}
       specs [block!] "Dialect block of layout"
       /local outstring
   11
       outstring: copy {}
       out: func [str /local pstr buffs][
           ;Break string into lines and output lines one at a time
           foreach lstr pstr: parse/all str "^/" [
               ;The ODROID-Show hardware is limited to 256 characters per
line, but this
                ; can be overcome by breaking the line into 256-character
chunks and
               ; outputting them one after the other.
               buffs: copy []
                ;Replace tabs with four spaces before breaking apart into
chunks.
               while [not tail? detab/size lstr 4][
                   append buffs copy/part lstr 256
                   lstr: skip lstr 256
               1
               foreach buff buffs [
                   write serport join buff "^J^M"
                     ; In my tests, I find that wait values less than 80ms
often lead
                     ; to output corruption. Longer waits may be required
with some
                   ; output.
                   wait .08
               1
               append outstring 1str
```

]]

```
font-rules: [
           color-rules (out join "^[[3" col)
           'default (out "^[[39m")
            | set font-sz integer! (out rejoin ["^[["
font-sz "s"])
           | set textout string! (out textout)
       1
       back-rules: [
           color-rules (out join "^[[4" col)
           'default (out "^[[49m")
       1
       color-rules: [
           'black (col: "0m")
           | 'red (col: "1m")
           | 'green (col: "2m")
           / 'yellow (col: "3m")
           | 'blue (col: "4m")
           | 'magenta (col: "5m")
           'cyan (col: "6m")
           / 'white (col: "7m")
       1
       cursor-rules: [
           'down (out "^[D")
           | 'up (out "^[M")
           'position? (out "^[[6n")
           'save (out "^[[s")
           'restore (out "^[[u")
           | 'to any [
              set cpos pair! (out rejoin ["^[[" cpos/x
";" cpos/y "H"])
               | 'home (out "^[[H")
               / 'bottom (out "^[E")
           1
       1
       ;'presses uses a fun trick. If keypresses have
been defined by the input from the user,
       ; then use that number. If not, use the default
```

; the old-school method that would look like this:

(1 in this case). This is shorter than

```
; unless value? 'keypresses [keypresses: 1]
; keypresses
;
; (52 bytes, not including the line break)
```

```
; ("unless" is a synonym in Rebol of "if not",
which is not any shorter than "unless")
```

```
; Another possible method could be like this:
             all [not value? 'keypresses keypresses:
        ;
1]
            keypresses
       :
       :
            (53 bytes, not including the line break)
       ;
              (The "do pick" method below is only 41
bytes)
       presses: [do pick [keypresses 1] value? 'key-
presses]
       key-rules: [
           'up (out rejoin ["^[[" do presses "A"])
              / 'down (out rejoin ["^[[" do presses
"B"])
              | 'left (out rejoin ["^[[" do presses
"C"])
              | 'right (out rejoin ["^[[" do presses
"D"])
           | set keypresses integer!
       1
       rotate-rules: [
           'left (loop 3 [out "^[[r"])
           / 'right (out "^[[r")
       1
       show-rules: [any [
           'font any font-rules
           | 'backdrop any back-rules
           / 'cursor any cursor-rules
           'reset (out "^[c")
           | 'clear (out "^[[2J")
           ' off (out "^[[0q")
           | 'on (out "^[[1q")
           / 'key any key-rules
           | 'rotate any rotate-rules
           | 'draw
           | set textout string! (out textout)
       ]]
       parse specs show-rules
       outstring
```

]

This simple Rebol program makes the ODROID-SHOW's commands more accessible by abstracting the native command syntax.

Please refer to the Rebol website for more information at http://www.rebol.com, or download the source code at http://www.rebolsource.net.

PACKAGE YOUR COMPILED SOFTWARE FOR INSTALLATION To install DOOM, type INSTALL and press ENTER **COMPILING DOOM**

PART 2 by Tobias Schaaf **DOOM** originally came on floppy disks that held only I.44MB of game data and were very fragile.

n the previous article, I outlined the process for compiling and porting software and games to the ODROID, using DOOM as an example. This month, I present a technique for packaging the compiled binaries for sharing and distribution.

Checkinstall

look like this:

Checkinstall is very useful as a simple way to create installation (.deb) files. It allows you to pack everything your program need into a single file and add required libraries as a dependency, so that you can install it on

systems that do not yet have the required libraries installed. If you run checkinstall on the sdldoom project, it will

root@odroid-wheezy:/home/odroid/sources/sdldoom-1.10\$ checkinstall

checkinstall 1.6.2, Copyright 2009 Felipe Eduardo Sanchez Diaz Duran This software is released under the GNU GPL.

The package documentation directory ./doc-pak does

not exist. Should I create a default set of package docs? [y]:

Preparing package documentation...OK

Please write a description for the package. End your description with an empty line or EOF. >> SDL Doom

```
**** Debian package creation selected ***
*********
```

This package will be built according to these values:

```
0 - Maintainer: [ root@odroid-wheezy ]
```

```
1 - Summary: [ SDL Doom ]
```

```
2 -
    Name:
             [ sdldoom ]
3 -
    Version: [ 1.10 ]
4 -
    Release: [ 1 ]
5 - License: [ GPL ]
             [ checkinstall ]
6 –
    Group:
7 - Architecture: [ armhf ]
8 - Source location: [ sdldoom-1.10 ]
9 - Alternate source location: [ ]
10 - Requires: [ ]
11 - Provides: [ sdldoom ]
12 - Conflicts: [ ]
13 - Replaces: [
```

Created by id Software ©1993 id Software, Inc. All Rights Reserved.

CH

Enter a number to change any of them or press ENTER to continue:

Checkinstall takes the name of the folder you are currently in as the package name (sdldoom) and automatically takes the -1.10 that was added to the folder name as the version number. If your folder names does not match, you can alter these names here as well as setting the right version here. Make sure to change #2 Name as well as #11 Provides.

Here you can configure your installation and add required packages such as libsdl1.2debian, which is the libsdl library needed for this build under #10 Requires. After pressing Enter, it should generate a nice little .deb file.



DOOM is where the fastest player with the biggest gun wins! Unless, of course, you've modified the source code to make yourself 100% invincible, or be able to shoot through walls.

Enter a number to change any of them or press ENTER to continue:

Installing with make install...

------Installation results _____ make[1]: Entering directory `/home/odroid/sources/ sdldoom-1.10' /bin/sh ./mkinstalldirs /usr/local/bin /usr/bin/install -c doom /usr/local/bin/doom make[1]: Nothing to be done for `install-data-am'. make[1]: Leaving directory `/home/odroid/sources/sdldoom-1.10' Installation successful _____ _____ Copying documentation directory... ./README.asm ./Changelog ./TODO ./FILES ./README.book ./README.SDL ./README.b ./README.gl Copying files to the temporary directory...OK Stripping ELF binaries and libraries...OK Compressing man pages...OK Building file list...OK Building Debian package...OK Installing Debian package...OK Erasing temporary files...OK Writing backup package...OK OK Deleting temp dir...OK **** Done. The new package has been installed and saved to /home/odroid/sources/sdldoom-1.10/sdldoom 1.10-1 armhf.deb You can remove it from your system anytime using: dpkg -r sdldoom *****

Although interactive mode is an easy method to create a deb file, another way is to give the parameters right when you start checkinstall, which allows for some extra options as well:

```
$ checkinstall --backup=no --install=no
--requires="libsdl1.2debian" --pkgname="sdldoom-
odroid" --pkgversion=1.10
```

If you start checkinstall without the "--backup=no" option, a tar file is going to be created with all the files that were copied during the install process. If you start checkinstall without the "--install=no" option, your software will be installed right after the package is done building. After all this is completed, you will find a nice little .deb file in your folder.

```
$ ls -1 *.deb
-rw-r--r- 1 root root 174490 Apr 5 15:16 sdldoom_1.10-1_
armhf.deb
```

The .deb file can then be copied and installed to any other ODROID, and if you have set the dependency requirements right, it should start right away. Getting the requirements correct will take some time and experience, depending on the project. The libsdl1.2-dev package actually only includes the header files for SDL (files ending on .h and located somewhere within /usr/ include/), but the actually SDL library is called libsdl1.2debian.

If you check on the libsdl1.2-dev package, you will see Depends: libsdl1.2debian (= 1.2.15-5)in the dependencies list.

```
apt-cache show libsdl1.2-dev
Package: libsdl1.2-dev
Source: libsdl1.2
Version: 1.2.15-5
Installed-Size: 2358
Maintainer: Debian SDL packages maintainers <pkg-sdl-
maintainers@lists.alioth.debian.org>
Architecture: armhf
Replaces: libsdl-dev
Provides: libsdl-dev
Depends: libsdl1.2debian (= 1.2.15-5), libasound2-dev,
libcaca-dev, libdirectfb-dev (>= 0.9.22), libglu1-me-
sa-dev | libglu-dev, libpulse-dev, libx11-dev, libts-
dev, libxext-dev
Conflicts: libsdl-dev
Description-en: Simple DirectMedia Layer development
files
SDL is a library that allows programs portable low
level access to a video
framebuffer, audio output, mouse, and keyboard.
This package contains the files needed to compile and
link programs which
use SDL.
Homepage: http://www.libsdl.org/
Description-md5: 9a82f59c5790721baad7ffc5f181d3d6
Tag: devel::library, role::devel-lib, uitoolkit::sdl
Section: libdevel
Priority: optional
                  pool/main/libs/libsdl1.2/libsdl1.2-
Filename:
dev_1.2.15-5_armhf.deb
Size: 861890
MD5sum: 4295708cab85d1eb546b449350dd2da6
SHA1: 76fa9923c9765d7b92e373df6fe12949f2092db5
SHA256: 71def7638b06e6711f6fa8d96724aa7eb238a7b10f9f-
b14192b5a5c1018d1322
```

LINUX GAMING

Libsdl1.2debian is the actual library that contains the binary files needed to run libsdl. As previously mentioned, the -dev file only includes the headers, and is useless if you only want to RUN the program. However, if you are unsure what libraries to add in checkinstall, you can add the -dev libs as well. This method is not as clean, but it will work well, since the program will always have the actual library as a dependency.

Some special notes about checkinstall

Since checkinstall copies the files to the location, you will most likely have to run it as root, since the normal user will not be allowed to copy files in certain folders and the build will fail.

Sometimes, checkinstall fails even though it has root privileges, and there sometimes seems to be a problem with the creation of folders when using checkinstall. If a program that you installed requires creating a folder, checkinstall may fail because it cannot create the folder to copy the files. To solve this, run make install first and then run checkinstall again.

Checkinstall is very helpful, as it will strip all binary files of unnecessary content such as comments and debug symbols that are no longer needed after the program is build completely, which can reduce the size of files greatly.

```
Size of the doom binary before stripping:

$ ls -lh doom

-rwxr-xr-x 1 root root 423K Apr 5 11:33 doom

Size of the doom library in the .deb file:

$ ls -lh doom

-rwxr-xr-x 1 root root 368K Apr 5 15:16 doom
```

The file size for DOOM might not make a very huge impact, but it can for other programs. For example, the kernel files provided by Hardkernel have a total of 354MB uncompressed.

```
$ du -h --max-depth=1
45M ./usr
307M ./lib
2.5M ./boot
354M .
```

And the same kernel files after stripping:

```
$ du -h --max-depth=1
45M ./usr
18M ./lib
2.5M ./boot
65M .
```

As we can see, the /usr and /boot directory have not changed at at all, since they have only header files or files that can not be excluded, but if you strip away the unnecessary comments symbols on the actual libraries in the /lib folder, the size is reduced from 307 to 18 MB, and still the libraries have



Some of the fiercest monsters in DOOM came straight from my worst nightmares. Meet Spider Demon, a 4-legged robot brain with a chaingun.

the same functionality as they did with 307 MB of size. As a result, the install size of the kernel will be reduced from 354 MB to 65 MB using checkinstall. Depending on the software packages you create, checkinstall can help you keep your software nice and clean (and slim).

Checkinstall won't always work. Some projects do not provide a make install function, and checkinstall won't work for those applications. But don't worry, as you get more experienced in compilation, you'll probably figure out how to build your own install routines.

Building a larger project

For DOOM, compiling and packaging is very simple, and we only had to download one file instead of ten, hundreds, or thousands of files, and there was only one view dependency.

As an example of a larger project, I included compilation instructions for OpenXCom. When you browse the OpenXcom homepage, click on the following link in the wiki sidebar: http://bit.ly/lrWH0ml.

Compiling OpenXCom

OpenXcom requires the following libraries to compile:

SDL (libsdl1.2) SDL_mixer (libsdl-mixer1.2) SDL_image (libsdl-image1.2) SDL_gfx (libsdl-gfx1.2), version 2.0.22 or later yaml-cpp, (libyaml-cpp), version 0.5 or later

This is great, because everything we need is right in front of us, mostly because of OpenXcom's excellent documentation. It's now a simple matter to install what we need.

apt-get install libsdl1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsdl-gfx1.2-dev

I skipped the yaml-cpp because if you compile it on Ubuntu, installing libyaml-cpp-dev will give you a version 0.5 or higher, but on Debian Wheezy, you only get libyamlcpp version 0.3 which doesn't work.

To overcome this, I compiled libyaml-cpp myself which can be installed by typing the following in Terminal:

```
$ wget http://oph.mdrjr.net/meveric/repository/liby-
aml-cpp0.5_0.5.1-1_armhf.deb
$ dpkg -i libyaml-cpp0.5_0.5.1-1_armhf.deb
```

Now that we have all the requirements, we can start to download the sources. On the top of the OpenXCom Wiki page, there is a link to a repository, which leads to a project page containing lots of folder and files. It will be very time-consuming to download them all with wget, but luckily there is a faster alternative.

From the git repository, we can download all of the files from the entire repository with just one command. First, get the URL of the project site:

```
$ git clone https://github.com/SupSuper/OpenXcom.git
Cloning into 'OpenXcom'...
remote: Reusing existing pack: 40755, done.
remote: Counting objects: 124, done.
remote: Compressing objects: 100% (118/118), done.
remote: Total 40879 (delta 67), reused 0 (delta 0)
Receiving objects: 100% (40879/40879), 14.05 MiB | 849
KiB/s, done.
Resolving deltas: 100% (33534/33534), done.
```

Well, that was nice, since we just downloaded about 700 files in a few seconds! For other projects that use subversion instead of git, the command line is very similar: svn checkout <url>
 Shown below is a list of the new files:

```
$ ls -1
total 152
                          82 Apr 5 17:12 autogen.sh
-rwxr-xr-x
           1 root root
drwxr-xr-x 3 root root 4096 Apr 5 17:12 bin
-rw-r--r-- 1 root root 32796 Apr 5 17:12 CHANGELOG.
txt
drwxr-xr-x 3 root root
                        4096 Apr 5 17:12 cmake
-rw-r--r-- 1 root root 5681 Apr 5 17:12 CMakeLists.
txt
-rw-r--r-- 1 root root 3385 Apr 5 17:12 configure.
ac
-rw-r--r-- 1 root root 35819 Apr 5 17:12 COPYING
drwxr-xr-x 2 root root
                        4096 Apr
                                  5 17:12 docs
                                  5 17:12 install
drwxr-xr-x
           6 root root
                        4096 Apr
drwxr-xr-x 2 root root
                        4096 Apr
                                  5 17:12 m4
           1 root root 21142 Apr
                                  5 17:12 Makefile.am
-rw-r--r--
                        4096 Apr
drwxr-xr-x 2 root root
                                  5 17:12 obj
                                  5 17:12 README.txt
-rw-r--r--
           1 root root
                        4446 Apr
drwxr-xr-x 5 root root
                        4096 Apr
                                  5 17:12 res
drwxr-xr-x 12 root root
                        4096 Apr
                                  5 17:12 src
```

This time, there is no configure and no Makefile either, just a Makefil.am. So what can we do now?

There are two things in this folder that will help: an executable file called: autogen.sh and a file called CMakeList.txt.

Autogen

Some projects have a file with the name autogen, automake or autoconf instead of a configure file in their source directory.

```
$ ./autogen.sh
aclocal: installing `m4/pkg.m4' from `/usr/share/aclo-
cal/pkg.m4'
autoreconf: Entering directory `.'
autoreconf: configure.ac: not using Gettext
autoreconf: running: aclocal --force -I m4
autoreconf: configure.ac: tracing
autoreconf: configure.ac: not using Libtool
autoreconf: running: /usr/bin/autoconf --force
autoreconf: configure.ac: not using Autoheader
autoreconf: running: automake --add-missing --force-
missing
configure.ac:106: installing `./config.guess'
configure.ac:106: installing `./config.sub'
configure.ac:17: installing `./install-sh'
configure.ac:17: installing `./missing'
Makefile.am: installing `./depcomp'
autoreconf: Leaving directory
```

The output is very short, and it's not apparent what has been done. Let's check if it did something.

```
$ ls -1
total 1204
-rw-r--r--
            1 root root
                         36830 Apr 5 17:23 aclocal.
m4
-rwxr-xr-x 1 root root
                             82 Apr
                                    5 17:12 autogen.
sh
drwxr-xr-x 2 root root
                          4096 Apr
                                   5 17:23 autom4te.
cache
drwxr-xr-x 3 root root
                          4096 Apr 5 17:12 bin
           1 root root
                        32796 Apr
                                   5 17:12 CHANGELOG.
-rw-r--r--
txt
drwxr-xr-x 3 root root
                          4096 Apr 5 17:12 cmake
-rw-r--r-- 1 root root
                         5681 Apr 5 17:12 CMakeLists.
txt
lrwxrwxrwx 1 root root
                           37 Apr 5 17:23 config.guess
-> /usr/share/automake-1.11/config.guess
                            35 Apr 5 17:23 config.sub
lrwxrwxrwx 1 root root
-> /usr/share/automake-1.11/config.sub
-rwxr-xr-x 1 root root 211749 Apr 5 17:23 configure
-rw-r--r--
           1 root root
                          3385 Apr
                                    5 17:12 configure.
ac
                                   5 17:12 COPYING
-rw-r--r-- 1 root root 35819 Apr
lrwxrwxrwx 1 root root
                                   5 17:23 depcomp ->
                            32 Apr
/usr/share/automake-1.11/depcomp
drwxr-xr-x 2 root root
                          4096 Apr
                                   5 17:23 docs
drwxr-xr-x
           6 root root
                          4096 Apr
                                   5 17:12 install
                           35 Apr 5 17:23 install-sh
lrwxrwxrwx 1 root root
-> /usr/share/automake-1.11/install-sh
drwxr-xr-x 2 root root
                          4096 Apr 5 17:23 m4
                         21142 Apr 5 17:12 Makefile.
-rw-r--r--
            1 root root
am
           1 root root 822542 Apr 5 17:23 Makefile.
-rw-r--r--
in
lrwxrwxrwx 1 root root
                            32 Apr 5 17:23 missing ->
/usr/share/automake-1.11/missing
drwxr-xr-x 2 root root
                          4096 Apr 5 17:12 obj
                           4446 Apr 5 17:12 README.
-rw-r--r--
            1 root root
txt
                          4096 Apr 5 17:12 res
drwxr-xr-x 5 root root
drwxr-xr-x 12 root root
                          4096 Apr 5 17:19 src
```

After this step, there is now a configure file, which gives the same starting point for compilation as the first example.

LINUX GAMING

CMakeLists.txt

Cmake is also a very nice tool, and if a project has a CMakeLists.txt in its directory, it supports cmake. We could just go ahead and type cmake, but i would highly advise NOT to do so!

Although cmake will work, and would generate the code that we need, it's somewhat messy. Also, you can't figure out what is source code and what comes from cmake, so there is a better way to do it.

Instead, create a new folder change in that folder and start cmake from there.

- \$ mkdir build
- \$ cd build
- \$ cmake ..

Cmake is started with a path to the CMakeLists.txt file. In the DOOM example, it was "cmake .". Since we were in the same directory we used the "." to tell cmake to use the current directory. Now that we are in a subfolder called build, we just tell cmake that the CMakeLists.txt is one folder up by typing "cmake ..". With everything setup correctly, it should look similar to this:

```
$ cmake ..
-- The C compiler identification is GNU 4.7.2
-- The CXX compiler identification is GNU 4.7.2
-- Check for working C compiler: /usr/bin/gcc-4.7
-- Check for working C compiler: /usr/bin/gcc-4.7 --
works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/g++-4.7
                                                            ſ
-- Check for working CXX compiler: /usr/bin/g++-4.7
-- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Looking for include file pthread.h
-- Looking for include file pthread.h - found
                                                            ſ
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
 - Found PkgConfig: /usr/bin/pkg-config (found version
"0.26")
-- Looking for XOpenDisplay in /usr/lib/arm-linux-
gnueabihf/libX11.so;/usr/lib/arm-linux-gnueabihf/
libXext.so
-- Looking for XOpenDisplay in /usr/lib/arm-linux-
gnueabihf/libX11.so;/usr/lib/arm-linux-gnueabihf/
libXext.so - found
-- Looking for gethostbyname
-- Looking for gethostbyname - found
-- Looking for connect
-- Looking for connect - found
-- Looking for remove
                                                            $
-- Looking for remove - found
  Looking for shmat
-- Looking for shmat - found
-- Looking for IceConnectionNumber in ICE
-- Looking for IceConnectionNumber in ICE - found
-- Found X11: /usr/lib/arm-linux-gnueabihf/libX11.so
-- Found OpenGL: /usr/local/lib/libGL.so
found SDL 1.2.15 (/usr/lib/arm-linux-gnueabihf:/usr/
```

include/SDL)

found SDL mixer 1.2.12 (/usr/lib/arm-linux-gnueabihf:/usr/include/SDL) found SDL_gfx 2.0.23 (/usr/lib/arm-linux-gnueabihf:/ usr/include/SDL) found SDL_image 1.2.12 (/usr/lib/arm-linux-gnueabihf:/usr/include/SDL) found yaml-cpp(/usr/local/lib:/usr/local/include/ yaml-cpp;/usr/local/include/yaml-cpp/..) -- Found Git: /usr/bin/git (found version "1.7.10.4") git found: /usr/bin/git version:0.9.g8bcafea No doxygen command found. Disable API documentation generation -- Configuring done -- Generating done -- Build files have been written to: /home/odroid/ sources/OpenXcom/build \$ ls -1 total 68 drwxr-xr-x 2 root root 4096 Apr 5 17:37 bin -rw-r--r-- 1 root root 29527 Apr 5 17:37 CMakeCache.

txt drwxr-xr-x 4 root root 4096 Apr 5 17:37 CMakeFiles -rw-r--r-- 1 root root 1906 Apr 5 17:37 cmake_install.cmake -rw-r--r-- 1 root root 3526 Apr 5 17:37 CPackConfig. cmake -rw-r--r-- 1 root root 3942 Apr 5 17:37 CPackSource-Config.cmake drwxr-xr-x 3 root root 4096 Apr 5 17:37 docs -rw-r--r-- 1 root root 6481 Apr 5 17:37 Makefile drwxr-xr-x 3 root root 4096 Apr 5 17:37 src

Since we got a Makefile here, so we run make and watch the pretty colors go by!

\$ make Scanning dependencies of target openxcom 0%] Building CXX object src/CMakeFiles/openxcom. dir/main.cpp.o 0%] Building CXX object src/CMakeFiles/openxcom. dir/lodepng.cpp.o 1%] Building CXX object src/CMakeFiles/openxcom. dir/Basescape/BaseView.cpp.o 1%] Building CXX object src/CMakeFiles/openxcom. dir/Basescape/CraftSoldiersState.cpp.o [...] [99%] Building CXX object src/CMakeFiles/openxcom. dir/Ufopaedia/ArticleStateTextImage.cpp.o [99%] Building CXX object src/CMakeFiles/openxcom. dir/Ufopaedia/ArticleStateArmor.cpp.o [100%] Building CXX object src/CMakeFiles/openxcom. dir/Ufopaedia/ArticleStateVehicle.cpp.o Linking CXX executable ../bin/openxcom [100%] Built target openxcom

Everything looks good, and we have an OpenxCom build in the bin folder. Now, run checkinstall again. We already know what libraries to add to the required list since we got that list from the OpenXCom site.

\$ checkinstall --backup=no --install=no --requires="libdl1.2debian, libsdl-image1.2, libsdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5" --pkgname="openxcom" --pkgversion="0.9.g8bcafea"

This is one of the moments when checkinstall will fail

because of its issue with creating directories, so run make	Installing: /usr/local/share/openxcom/data/Re-
install first and then re-run checkinstall.	Installing: /usr/local/share/openxcom/data/Re-
<pre>\$ checkinstallbackup=noinstall=no requires="libdl1.2debian, libsdl-image1.2, lib- sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5" pkgname="openxcom"pkgversion="0.9.g8bcafea" checkinstall 1.6.2, Copyright 2009 Felipe Eduardo San- chez Diaz Duran This software is released under the GNU GPL.</pre>	 - Installing: /usr/local/share/openxcom/data/Shaders - Installing: /usr/local/share/openxcom/data/Shaders/Phosphor-simple.OpenGL.shader - Installing: /usr/local/share/openxcom/data/Shaders/Openxcom.OpenGL.shader CMake Error at src/cmake_install.cmake:54 (FILE): file INSTALL cannot find "/home/odroid/sources/test/OpenXcom/bin/data/Shaders/heavybloom.OpenGL.shader". Call Stack (most recent call first):
The package documentation directory ./doc-pak does not	<pre>cmake_install.cmake:38 (INCLUDE)</pre>
exist. Should I create a default set of package docs? [y]:	<pre>make: *** [install] Error 1</pre>
Y	<pre>**** Installation failed. Aborting package cre- ation.</pre>
Please write a description for the package. End your description with an empty line or EOF. >> OpenXCom	Cleaning upOK
»	Bye.
***** Debian package creation selected ***	
*****	\$ make install
This package will be built according to these values.	[100%] Built target openxcom
This package will be built accoluting to these values.	Install the project
0 - Maintainer: [root@odroid-wheezy]	Installing: /usr/local/bin/openxcom
1 - Summary: [OpenXCom] 2 - Name: [openxcom]	Removed runtime path from "/usr/local/bin/openx-
3 - Version: [0.9.g8bcafea]	com"
4 - Release: [1]	Installing: /usr/local/share/openxcom/data/Re-
5 - License: [GPL] 6 - Group: [checkinstal]]	sources
7 - Architecture: [armhf]	 Installing, /usr/logal/sharo/oponygom/data/Bulosot
8 - Source location: [build]	Installing: /usr/local/share/openxcom/data/Rule-
9 - Alternate source location: [] 10 - Requires: [libdl1.2debian, libsdl-image1.2, lib-	set/Xcom1Ruleset.rul
sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5]	
11 - Provides: [openxcom]	
12 - Confficts: [] 13 - Replaces: []	<pre>\$ checkinstallbackup=noinstall=no</pre>
Enter a number to change any of them or press ENTER to continue:	<pre>requires="libdl1.2debian, libsdl-image1.2, lib- sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5" pkgname="openxcom"pkgversion="0.9.g8bcafea"</pre>
Installing with make install	checkinstall 1.6.2, Copyright 2009 Felipe Eduardo San- chez Diaz Duran
Installation results	This software is released under the GNU GPL.
[100%] Built target openxcom	******
Install the project	**** Debian package creation selected ***
Install configuration: ""	****
Installing: /usr/local/bin/openxcom Removed runtime path from "/usr/local/bin/openx- com"	This package will be built according to these values:
Installing: /usr/local/share/openxcom/data	0 - Maintainer: [root@odroid-wheezy]
Installing: /usr/local/share/openxcom/data/Re-	1 - Summary: [OpenXCom] 2 - Name: [openXcom]
Installing: /usr/local/share/openxcom/data/Re-	3 - Version: [0.9.g8bcafea]
sources/UI	4 - Release: [1]
Installing: /usr/local/share/openxcom/data/Resourc-	5 - License: [GPL] 6 - Group: [checkinstal]]
Installing: /usr/local/share/openxcom/data/Re-	7 - Architecture: [armhf]
sources/Weapons	8 - Source location: [build]
Installing: /usr/local/share/openxcom/data/Resourc-	<pre>y - Alternate source location: [] 10 - Requires: [libdl1.2debian, libsdl-image1.2, lib-</pre>
Installing: /usr/local/share/openxcom/data/Resourc-	sdl-mixer1.2, libsdl-gfx1.2-4, libyaml-cpp0.5]
es/Pathfinding	11 - Provides: [openxcom]
Installing: /usr/local/share/openxcom/data/Resourc-	12 - CONNICTS: [] 13 - Replaces: []
Installing: /usr/local/share/openxcom/data/Resourc-	
es/BulletSprites	Enter a number to change any of them or press ENTER
Installing: /usr/local/share/openxcom/data/Resourc-	to continue:
er, Farreer, Farreer, Fuid	

LINUX GAMING

```
Installing with make install ...
                          Installation
                                         results
_____
_____
[100%] Built target openxcom
Install the project...
-- Install configuration: ""
-- Up-to-date: /usr/local/bin/openxcom
-- Installing: /usr/local/share/openxcom/data
-- Installing: /usr/local/share/openxcom/data/Resourc-
es
. . .
-- Installing: /usr/local/share/openxcom/data/Ruleset
-- Up-to-date: /usr/local/share/openxcom/data/Ruleset/
Xcom1Ruleset.rul
_____
                        Installation
                                      successful
_____
Copying documentation directory...
./
./docs/
./docs/Makefile
./docs/CMakeFiles/
./docs/CMakeFiles/CMakeDirectoryInformation.cmake
./docs/CMakeFiles/progress.marks
./docs/cmake_install.cmake
Some of the files created by the installation are inside
the home directory: /home
You probably don't want them to be included in the
package.
Do you want me to list them? [n]: n
Should I exclude them from the package? (Saying yes is
a good idea) [n]: y
Copying files to the temporary directory...OK
Stripping ELF binaries and libraries...OK
Compressing man pages...OK
Building file list...OK
Building Debian package...OK
NOTE: The package will not be installed
Erasing temporary files...OK
Deleting temp dir...OK
*****
Done. The new package has been saved to
/home/odroid/sources/test/OpenXcom/build/
openxcom_0.9.g8bcafea-1_armhf.deb
You can install it in your system anytime using:
   dpkg -i openxcom_0.9.g8bcafea-1 armhf.deb
```

One this step completes, the OpenXcom .deb installation package is ready for use! This should be enough information to get started with compiling your first projects.

Notes on cmake

There is something that I rather like about cmake. After cmake is done creating the build-tree for a project, you will find a folder with the name of the binary file of the project. For example, on the OpenXcom project, the folder can be found at src/CMakeFiles/openxcom.dir/. In this folder, you will always find a file called link.txt which will show you how the final binary is built as well as which libraries are linked into it.



The Cyber Demon has a frightening description in the original DOOM manual: "a missile-launching skyscraper with goat legs. 'Nuff said".

\$ cat src/CMakeFiles/openxcom.dir/link.txt
[...]

-o ../bin/openxcom -rdynamic -lSDL_image -lSDL_mixer -lSDL_gfx -Wl,-Bstatic -lSDLmain -Wl,-Bdynamic -lSDL -lpthread /usr/local/lib/libyaml-cpp.so /usr/local/ lib/libGL.so -Wl,-rpath,/usr/local/lib:

If you check the last line, you will find the libraries that are linked to the binary file, for example, -lSDL_image -lSDL_ mixer -lSDL_gfx -lSDL. As mentioned in the previous article, the files starting with "-l" are libraries. These are exactly the libraries that we used in our checkinstall command, and that were listed on the OpenXCom site. So, in case you don't know which libraries you needed, you can also find them here.

Checking dependencies

With the 1dd command, you can easily check on the dependencies of a particular binary file that has already been compiled.

```
$ ldd /usr/local/bin/doom
libm.so.6 => /lib/arm-linux-gnueabihf/libm.so.6
(0xb6e80000)
libSDL-1.2.so.0 => /usr/lib/arm-linux-gnueabihf/libS-
DL-1.2.so.0 (0xb6e03000)
libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6
(0xb6d1e000)
/lib/ld-linux-armhf.so.3 (0xb6f08000)
[...]
```



ODROID MAGAZINE 16

DESCRIBING THE MATHEMATICAL FUNCTION ATAN2 A USEFUL TOOL FOR PROGRAMMING APPLICATIONS THAT REQUIRE REAL-TIME TRIGONOMETRY

by Jussi Opas

any of us have been applying the casual trigonometric functions such as sin, cos or tan and their reverse functions asin, acos, and atan to make geometric computations. That is what also school mathematics has taught us to do. While doing so, one meets such problems as, the value interval of sin is $[-\pi/2, \pi/2]$, and there is a chance to divide with zero, when $\tan(y/x)$ is used. To solve these problems, programs must branch with "if" sentences, and the division by zero problem must be handled, too. There is an excellent method for solving all the mentioned problems within computer programs by applying the atan2(y, x) method.

The atan2 function was originally introduced in FOR-TRAN programming language to ease up trigonometric computations [WIK14]. It is available also elsewhere, like C, Java, and (surprise, surprise) in .NET and Excel, too. The advantage of atan2 is that it gives continuous values in the interval of $(-\pi, \pi]$. When expressed with atan function, the definition is as follows:

```
atan2(y,x)=atan(yx),ifx>0atan(yx)+\pi,ify>0,x<0atan(yx)-\pi,ify<0,x=0\pi2,ify>0,x=0-\pi2,ify<0,x=0undefined,ify=0,x=0
```

When one looks at the formula, the conclusion is that it is quite a complex function, with many branches to implement the functionality of atan2 with atan instead. The same applies also if one tries to use asin or acos. An issue with the above definition is that there is no defined value when both x and y



are zero. Another note is that the order of parameters is y, x instead of x, y. But, it is in reverse order in Excel.

> In a mouse-interactive clock, we can compute, from the mouse location, the angle of selected pointer to determine how the selected time can be adjusted. For instance, one can calculate the angle from north to mouse click as follows:

double angleAsRadians = Math.atan2(point.y - centerY, point.x - centerX); // atan2 returns values in the range of [-PI, PI] // so, we can get values into the ranges [0, 2PI] by adding one PI angleAsRadians += PI;

It is just so awesome that no "if" statements are needed. Though, this is not expensive like a shooting game, where the angle and distance to target must be calculated continuously. Trigonometry is also heavily used in ray tracing applications, such as radio coverage and 3D modelling, where atan2 can be successfully applied. So, atan2 is a beast in its expressiveness, therefore we can safely call it the best trigonometrical function in a programmer's toolkit.

The behavior of atan2 can be shown most easily with some mathematics software such as Octave or Maxima. Both of them are available in Ubuntu Software Center. Android has its own versions available in PlayStore.



In the plot window, one can rotate to an appropriate view angle with the mouse and watch the 3 dimensional view from an appropriate angle.

Software development is driven by use cases and by user

stories to be fulfilled. Projects are often very busy with implementing all necessary functionality, and performance testing is done only in the end of the project. At that time it may be even too late to react on poor performance, since it may be difficult to improve in late project phase. Therefore it would be nice to already know what the performance implication will be at the program writing time.

If an application uses intensive math like distance and angular computations, it would be good to know what it costs to call these methods. In Java, for instance, calls like Math.sqrt, Math.atan, or Math.atan2 are then important to understand.

We made a test by calling different distance and angle computations methods 10 million times with Java. The test framework adds on method call overhead to each functionusing an ODROID XU at 1.6 GHz frequency running Ubuntu with Java. The relative behavior of the tested math methods are similar in other platforms. The results are shown in the enclosed graph.



Computation of distance is fast, that is calling of sqrt(x*x + y*y)*r takes only 24 nano seconds on average. Multiplication with resolution r is almost free, since calling of plain sqrt(x*x, y*y) is only 4 nanoseconds faster. Then, one sees clearly that calling of pow(x,2) is essentially more expensive than x*x. The next observation is that angle computation methods are slow, especially Java's trigonometric functions, which is a tradeoff for their accurateness. In this case, the precious atan2 seems to be slower that atan.

If we want to use the atan2 method, we must find some means to compute it more quickly. We could ourselves implement a faster, but less accurate, method to compute atan2. We could use parabolic approximation, Taylor series or Chebyshev series or similar, [COR09], or a software implementation of the famous CORDIC method. On the other hand, we could implement some table lookup method. In the first place, utilization of prior computed values might be the easiest to implement by creating the following look up methods for fast access of atan2 values:

- binary search based on Java's Arrays.binarySearch method
- binary find is a self implemented method to find aimed value from array
- table lookup with a large n*n table, and, an
- octet math is another table lookup method where each symmetric octet of a unit circle is represented as an array, [RIV06].

The next figure shows the call time of atan2 against the used look up methods. In the representative run, it took 368 nanoseconds to compute. Lookup methods are compared against that value. The call of one access method is shown with blue bar and the achieved improvement has been shown with orange gradient bar.

It is a surprise that a self written binary find is faster than the binary search that is based on Java's standard method. One should be suspicious of that, and the result would need to be verified prior to using a newly written method in production code.

The two array based look up methods, table lookup and octet math are the fastest. By using octet math, an atan2 value can be completed in about 50 nanoseconds. That is the method that we have been using in our sample application, too.



Here is the source code for those who are interested in the actual Java implementation:

/**
* Octet based atan2 lookup method.
* Created by xyz on 12/13/13.
*/
<pre>public class OctetMath {</pre>
private static final double ROUND = 0.5;
<pre>private static final double PI = Math.PI;</pre>
<pre>// To be accurate enough, at least 2000 values are</pre>
needed, we use exponent of two in here.
// The memory consumption is $2048 * 8 * 4 = 65536$
bytes ~ 65.5 kBytes.
<pre>// Using of bigger size will increase accuracy.</pre>
private static final int SIZE = 2048;
// last index per side
private static final float LAST = SIZE - 1;
private static final float NEGATIVE_LAST = -LAST;
<pre>// Octets are illustrated below.</pre>
// ^ у
// 8 1st octet
// 7 2nd octet

```
11
               5 İ
                   4
    11
    // Array representations of strictly computed atan2
values.
    // Using double would not make the result more ac-
curate.
        private static final float[] OCTET1 =
                                                     new
float[SIZE];
                          final
                                 float[]
                                          OCTET2
        private
                  static
                                                  =
                                                      new
float[SIZE];
        private
                  static
                          final
                                 float[]
                                          OCTET3
                                                  =
                                                      new
float[SIZE];
                  static
                          final
                                 float[]
                                          OCTET4
        private
                                                  =
                                                      new
float[SIZE];
        private
                          final
                                 float[]
                  static
                                          OCTET5
                                                     new
float(SIZE):
                          final
                                 float[]
                                          OCTET6
                  static
        private
                                                  =
                                                     new
float[SIZE];
                  static
                          final
                                 float[]
                                          OCTET7
        private
                                                     new
float[SIZE];
        private
                  static final float[]
                                         OCTET8
                                                     new
float[SIZE];
    static {
        initializeOctets(SIZE);
    }
    /**
     * Get atan2 as table lookup.
      * The complexity of the method is 14, but we do
not want this method to be split.
     * @param y
     * @param x
      * @return angle as radians, the value is in the
interval [-PI, PI]
    @SuppressWarnings("all")
     public static float atan2(final float y, final float
x) {
        if (0 <= x) {
             if (0 <= y) {
                 if (x < y) {
                      return OCTET1[(int) (LAST * x / y
+ ROUND)];
                 } else {
                      return OCTET2[(int) (LAST * y / x
+ ROUND)];
                 }
             } else {
                 if
                    (-y < x) {
                     return OCTET3[(int) (NEGATIVE_LAST
 y / x + ROUND];
                 } else {
                     return OCTET4[(int) (NEGATIVE_LAST
* x / y + ROUND)];
                 }
             3
        } else {
             if (y < 0) {
                 // both x and y are negative
                 if (x > y) {
                      return OCTET5[(int) (LAST * x / y
+ ROUND)];
                 } else {
                      return OCTET6[(int) (LAST * y / x
+ ROUND)];
                 }
             } else {
                 if (y < -x) {
                     return OCTET7[(int) (NEGATIVE_LAST
      x + ROUND)];
                 } else {
                     return OCTET8[(int) (NEGATIVE LAST
* x / y + ROUND)];
```

---> x

3

11

11

6

```
}
    }
    private static void initializeOctets(final double
size) {
        final double y = size;
        for (double i = 0; i < size; i++) {
            final double x = i;
            final double theta = Math.atan2(y, x);
            final int index = (int) i;
            OCTET1[index] = (float) theta;
              OCTET2[index] = (float) (PI / 2.0 - the-
ta);
             OCTET3[index] = (float) (-PI / 2.0 + the-
ta);
            OCTET4[index] = (float) -theta;
            OCTET5[index] = (float) (-PI + theta);
             OCTET6[index] = (float) (-PI / 2.0 - the-
ta);
              OCTET7[index] = (float) (PI / 2.0 + the-
ta);
            OCTET8[index] = (float) (PI - theta);
        }
    }
    private OctetMath() {
        11
    3
}
```

One should always verify self-made implementations, since the accuracy of the method is of concern. The correctness and reached accuracy can be tested against the built-in Math. atan2 Java method. The accuracy can be improved by using larger arrays to represent prior computed atan2 values. The disadvantage of bigger arrays is, of course, that more statically allocated memory is needed. The memory consumption of three different memory layouts is shown comparatively in logarithmic scale in the next figure.



Array and Memory Size

The table memory layout is consuming the most memory. The worst thing is that its behavior is exponential. As the 8 octets of a unit circle are symmetric, the same method could be implemented by storing only the data of one of the octets, in which case only 1/8 of the memory amount is needed. Then the call time would be bigger, since more control flow branches must be traversed during one call.

Summary

With an ODROID, one can make mathematical studies with dedicated tools, by implementing his own application and studying its behavior. We warmly recommend application of atan2 in computer programs, especially intensive programs that use a lot of angle and distance computations such games and ray tracing. Java's accurate trigonometric methods are slow. Therefore some faster methods should be available. We implemented an octet based table lookup method in order to access atan2 values faster. Our self-made implementation OctetMath.atan2 is about 7x faster than the standard Math.atan2. It allocates 66 kB of static memory, which is affordable in terms of resources.

References

[WIK14] atan2. 8 pages, 2014. http:// en.wikipedia.org/wiki/Atan2 [COR09] Coranac / Vijn Jasper. Another fast fixed-point sine approximation. 16.7.2009. www.coranac.com/2009/07/ sines [RIV06] Riven X. 13.8x faster atan2. 2006. http://www.java-gaming.org/topics/13-8x-faster-atan2updated/14647/view.html

FRAMEBUFFER TERMINAL CONSOLE FOR THOSE GUI-LESS MOMENTS

by Rob Roy

hen running an Ubuntu distribution running Kernel 3.8.13 or above on an ODROID-X/X2/U2/U3, there is a resident framebuffer terminal console that can be activated by pressing Control-Alt-F[1-6]. To return to the X11 graphical interface, press Control-Alt-F7. This gives you access to the command line even if the X11 interface has stopped working.

TECHNICAL ARTICLE (BEGINNER)

INSTALLING MATHEMATICAL TOOLS FROM THE UBUNTU SOFTWARE CENTER CREATE BEAUTIFUL 3D GRAPHS FOR YOUR OFFICE AND IMPRESS YOUR COLLEAGUES

by Jussi Opas

he Ubuntu Software center contains many free and low-priced applications, including scientific and mathematical programs. Not all of them are available for ARM processor based computers, but many of them run natively on the ODROID, and the Software Center ensures that one can load a correctly compiled version. This article describes how to load a mathematics application from the Ubuntu store.

In the upper right corner of the Ubuntu Software Center, there is a search box which can be used if we know approximately what we are looking for. In the left there is a category list, as shown in the first screenshot.

Accessories
Books & Magazines
Developer Tools
Education
Fonts
Games
Graphics
Internet
Office
Science & Engineering
Sound & Video
System
Themes & Tweaks
Universal Access

To find a mathematics tool, push the 'Science & Engineering' button. Then the store gives a list of alternatives represented as icons.



Select the Mathematics icon, and the next window will show a collection of tools that are useful for doing mathematics. It is easy to install a selected tool, and

Ubuntu Software Center IIC Edit View Help	- + ×	*	Plot 3D		×
All Software - Latalied - History Progress		Expression	atan2(y,x)		
Science & Engineering Mathematics By 10	op Rated 💌	Variable:	x From: -5	To: 5	
GNU Octave ***** (14) Scientific Computing using GNU Octave		Variable:	V From: 5	To: 5	
R ****** (6) R software environment for statistical computing and graphics		variable.	y 110115	10. 5	
Multi Kmplot ***** (22) Function Plotter		Grid:	30 ‡ x 30	*	
Tux Math A and Mith Tus		Format:	gnuplot 👻		
wooMaxima ***** (36) Perform symbolic and numeric calculations using Maxima				J	
MATLAD ***** (38) Scientific computing environment		Options:		*	ry pm3d
Gnumeric Spreadsheet ***** (54) Calculation, Analysis, and Visualization of Information		Plot to file:			
QLOCLave ***** (21) Graphical Qt frontend for GNU Octave					
Fraqtive ***** (15) Show 487.technical items				Cancel	√ ОК
Mathematics Ubuntu Shon		Plot 3D			
manomanos osanta onop		1 101 00			

uninstallation is also possible from this interface.

To start, I recommend loading wxMaxima and/or QtOctave. To demonstrate that the programs are working, let's paint atan2 with Maxima and Octave.

After installation, wxMaxima appears in the Education menu. Select 'Plot 3D...' from wxMaxima's Plot menu to see the plotting dialog.

Next, write your favorite mathematical formula into the Expression field and fill in the value intervals for x and y.



In the plot window, one can rotate to an appropriate viewing angle and see the 3D illustration from any side. This same can be written also into a wxMaxima applications command line as follows: plot3d(atan2(y,x), [x,-5,5], [y,-5,5], [plot_ format,gnuplot], [gnuplot_pm3d,true]).

Also in QtOctave, one can define a formula, and then rotate the resulting image with a mouse.

Mathematical tools are available also on Android in the Play Store. The Maxima Android version is shown in the next figure. However, in Android, one can not rotate the picture, unlike the Linux version of the Maxima application.



Because Ubuntu is popular for running scientific applications, there are many interesting plotting programs available to verify your experiment results, visualize mathematical formulas, and create intricate 3D graphs. There is also a dedicated engineering and scientific group on the official Ubuntu Engineering site at http://bit.ly/lvVlVHx, where you can find CAD programs, electronics applications, data visualization packages, and much more to help you have fun with your mathematical projects on the ODROID!

QTOctave Atan2

Mathematica





ANDROID IMAGE FILES A PEEK INTO THE COMPRESSED FILES THAT MAKE ANDROID PORTABLE AND LIGHTWEIGHT

by Nanik Tolaram

he Android source code is incredibly massive (more than 4GB), but how does this large code base gets installed onto phones, tablets and our ODROID boards? The simple answer is that the source gets compiled into compact binary files and is then compressed to a particular format. In the last 2 issues, we have been looking into Android source and how to compile it for ODROID-U3. In this month's article, we are going to explore the different image files that are generated by the Android build system. Please note that all software tools used in this article can be downloaded from http://bit.ly/1hy3vdc.

Android devices run on very limited resources and disk space, which makes it crucial that the operating system, along with the supporting files, be as small as possible. The final image files generated by the build system are:

```
boot.img
cache.img
recovery.img
system.img
userdata.img
```

There are a few scenarios in which knowing more about the image files may benefit:

Removing or adding files to the image file Experimenting with different configurations quickly Upgrading Android versions without compiling the full source code



boot.img

This image file contains the initialization (init) file that is used to boot up Android. The file init.rc contains information for hosting the different services and configurations that are needed to start Android. There are 2 files inside the /sbin directory, and one in particular is very important, called adbd, which is the adb daemon that is run as part of the Android startup process. Adbd allows you to connect to your device using the adb command.

The following steps shows you how to extract the boot.img:

1. Run the unmkbooting utility by typing

```
$ unmkbooting boot.img.
Kernel size 3133124
Kernel address 0x10008000
Ramdisk size 167690
Ramdisk address 0x11000000
Secondary size 0
Secondary address 0x10f00000
Kernel tags address 0x10000100
Flash page size 2048
Board name is ""
Command line ""
Extracting kernel.gz ...
Extracting initramfs.cpio.gz ...
All done.
_____
To recompile this image, use:
mkbooting --kernel kernel.gz --ramdisk initramfs.cpio.gz -o new boot.img
```

ODROID MAGAZINE 23

MULTIBOOT YOUR ODROID

A set of directories files are then placed into the current directory.

total 6460								
drwxrwxr-x	2	nanik	nanik	4096	May	10	22:09	
drwxrwxr-x	14	nanik	nanik	4096	May	10	21:53	
- FW- F F	1	nanik	nanik	3303424	May	10	22:09	boot.img
- FW- FW- F	1	nanik	nanik	167690	May	10	22:09	initramfs.cpio.gz
Google Chro	me	nanik	nanik	3133124	May	10	22:09	kernel.gz

File list after extracting boot.img

2. Run the extract_ulmage.sh script by typing

\$ extract uImage.sh initramfs.cpio.gz. Checking for uImage magic word... 1+0 records in 0+1 records out 4 bytes (4 B) copied, 3.9012e-05 s, 103 kB/s uImage recognized. Extracting data ... 2619+1 records in 327+1 records out 167626 bytes (168 kB) copied, 0.00185786 s, 90.2 MB/s Checking for ARM mach-type ... 3+0 records in 0+1 records out 3 bytes (3 B) copied, 3.5863e-05 s, 83.7 kB/s Checking for zImage... 1+0 records in 0+1 records out 4 bytes (4 B) copied, 2.6783e-05 s, 149 kB/s >>> initramfs.cpio.gz extracted to Image

Which the results in the files show here:

total 6624									
drwxrwxr-x	2	nanik	nanik	4096	May	10	22:11	./	
drwxrwxr-x	14	nanik	nanik	4096	May	10	21:53	/	
- FW- F F	1	nanik	nanik	3303424	May	10	22:09	boot.img	
- FW- FW- F	1	nanik	nanik	167626	May	10	22:11	Image.gz	
- FW- FW- F	1	nanik	nanik	167690	May	10	22:09	initramfs.cpio.gz	
- FW- FW- F	1	nanik	nanik	3133124	May	10	22:09	kernel.gz	

File list after extracting initramfs.cpio.gz

3. The extracted file is now a gzip file, so we need to rename it.

\$ mv Image Image.gz

4. Extract the newly renamed file using gunzip.

\$ gunzip Image.gz

Check the file type using the file tool by typing

\$ file Image

Image: ASCII cpio archive (SVR4 with no CRC).

5. Use the cpio tool to extract the content

\$ cpio -i < ./Image</pre>

563 blocks.

You will now see the full contents of the boot.img as shown below. Previous files such as boot.img have been removed from the screenshot for easier viewing.

Complete file list of boot.img

cache.img

This image file does not contain anything major. As the name implies, it is used as cache storage. On an Android device, this

particular image file will reside in its own partition, or sometimes its own directory called /cache. The cache is used by Android OTA (Over The Air) application to store updated image files, or to upgrade your Android system.

The following steps shows you how to extract the cache. img:

I. Use the simg2img tool to extract the image file by typing

\$ simg2img cache.img cache.raw

computed crc32 of 0xc76ce614, expected 0x0000000

2. Using command file cache.raw you will see the type of the extracted file.

\$ file cache.raw cache.raw: Linux rev 1.0 ext4 filesystem data, UUID=57f8f4bc-abf4-655f-bf67-946fc0f9f25b (extents) (large files)

3. Create a directory to mount the extracted file to the directory.

\$ mkdir cc \$ sudo mount -t ext4 -o loop cache.raw cc/

4. You will see there is nothing inside the cc/ directory.

total 8	-		100000	1005	400			
drwxr-xr-x	2	root	root	4096	Jan	1	1970	
drwxrwxr-x	3	nanik	nanik	4096	May	10	22:26	
Complete file	e lis	st of cach	e.img					

recovery.img

The recovery image contains Linux kernel and busybox tool, and resides in its own partition. In normal Android devices, there is a particular physical key sequence that you normally have to press in order to instruct the device to boot to the recovery partition. There is another way to boot to recovery using the following command inside Android:

adb reboot recovery



Follow these steps to extract the recovery.img:

1. Run the unmkbooting utility

```
$ unmkbooting recovery.img
Kernel size 3133124
Kernel address 0x10008000
Ramdisk size 1388735
Ramdisk address 0x11000000
Secondary size 0
Secondary address 0x10f00000
Kernel tags address 0x10000100
Flash page size 2048
Board name is ""
Command line ""
Extracting kernel.gz ...
Extracting initramfs.cpio.gz ...
All done.
_____
```

To recompile this image, use:

mkbooting --kernel kernel.gz --ramdisk initramfs. cpio.gz -o new_boot.img

total 8848								
drwxrwxr-x	2	nanik	nanik	4096	May	10	22:34	./
drwxrwxr-x	14	nanik	nanik	4096	May	10	21:53	/
- FW- FW- F	1	nanik	nanik	1388735	May	10	22:34	initramfs.cpio.gz
- rw- rw- r	1	nanik	nanik	3133124	May	10	22:34	kernel.gz
-rw-rr	1	nanik	nanik	4526080	May	10	22:34	recovery.img

```
Extracted file list for recovery.img
```

By using the file command to the check the type of the initramfs.cpio.gz file, you will see the following:

\$ file

./initramfs.cpio.gz: u-boot legacy uImage, ramdisk, Linux/ARM, RAMDisk Image (Not compressed), 1388671 bytes, 2 10:35:12 2014, Load Address: 0x40800000, Wed Apr Entry Point: 0x40800000, Header CRC: 0x70CA98DA, Data CRC: 0xC14A4AFD

2. Run the extract_ulmage.sh script

```
$ extract_uImage.sh initramfs.cpio.gz
Checking for uImage magic word ...
1+0 records in
0+1 records out
4 bytes (4 B) copied, 2.322e-05 s, 172 kB/s
uImage recognized.
Extracting data ...
21697+1 records in
2712+1 records out
1388671 bytes (1.4 MB) copied, 0.0140089 s, 99.1 MB/s
Checking for ARM mach-type ...
3+0 records in
0+1 records out
3 bytes (3 B) copied, 3.6532e-05 s, 82.1 kB/s
Checking for zImage...
1+0 records in
0+1 records out
4 bytes (4 B) copied, 3.2578e-05 s, 123 kB/s
>>> initramfs.cpio.gz extracted to Image
```

3. The extracted file is now a gzip file, so we need to rename it:

\$ mv Image Image.gz

\$ gunzip Image.gz

4. Extract the newly renamed file using gunzip:

np	0 Items folder	🖝 🔚 dev	0 items folder
(Empty)		(=	
stem	1 item folder	(Empty)	
bin	0 items folder	🔻 🚞 data	0 items folder
(Empty)		(=)	
15 (F=======)	0 items folder	(Empty)	
(Empty)	185 items folder	ueventd.rc	3.9 kB plain text document
s	2 Items folder		
mages	21 items folder		1.7 KB plain text document
stitch.png	162.2 kB PNG image	ueventd.goldfish.rc	272 bytes plain text document
progress_fill.png	404 bytes PNG image		
- progress_empty	118 bytes PNG image	init.rc	1.4 kB plain text document
- indeterminate06	676 bytes PNG image	init	105.2 kB executable
indeterminate05	683 bytes PNG image	V IIIIC	TOJ.2 KD EXECUTABLE
indeterminate04	665 bytes PNG image	default.prop	1.9 kB plain text document
- Indeterminate03	661 bytes PNG Image		
 Indeterminate03 indeterminate02 	661 bytes PNG Image 687 bytes PNG image		
 Indeterminate03 Indeterminate02 indeterminate01 	661 bytes PNG image 687 bytes PNG image 673 bytes PNG image		
- Indeterminate03 - Indeterminate02 - indeterminate01 - icon_installing_o	661 bytes PNG image 687 bytes PNG image 673 bytes PNG image 24.5 kB PNG image		
icon_installing_o	661 bytes PNG image 687 bytes PNG image 673 bytes PNG image 24.5 kB PNG image 23.9 kB PNG image		
Indeterminate03 Indeterminate02 Indeterminate01 Icon_installing_0 Icon_installing_0 Icon_installing_0	661 bytes PNG Image 687 bytes PNG Image 673 bytes PNG image 24.5 kB PNG image 23.9 kB PNG image 23.8 kB PNG image		
 indeterminateo3 indeterminateo2 indeterminateo1 icon_installing_o icon_installing_o icon_installing_o icon_installing_o 	661 bytes PNG image 687 bytes PNG image 673 bytes PNG image 24.5 kB PNG image 23.9 kB PNG image 23.8 kB PNG image 24.1 kD PNG image	Complete file liet	for recovery ind
Indeterminate03 Indeterminate02 Indeterminate01 Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o	661 bytes PNG image 687 bytes PNG image 673 bytes PNG image 24.5 kB PNG image 23.8 kB PNG image 24.1 kD PNG image 23.8 kB PNG image	Complete file list	for recovery.img
Indeterminate03 Indeterminate01 Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o	601 bytes PNG image 687 bytes PNG image 673 bytes PNG image 2.8.5 kB PNG image 2.8.5 kB PNG image 2.8.1 kB PNG image 2.8.5 kB PNG image 2.4.1 kB PNG image	Complete file list	for recovery.img
Indeterminateo3 Indeterminateo2 Indeterminateo1 Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o	60 bytes PNG image 673 bytes PNG image 73 bytes PNG image 23.9 kB PNG image 23.4 kB PNG image 24.1 kD PNG image 24.1 kD PNG image 24.4 kB PNG image 24.2 kB PNG image	Complete file list	for recovery.img
Indeterminateo3 Indeterminateo1 Indeterminateo1 Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o	661 bytes PNG image 673 bytes PNG image 673 bytes PNG image 23.5 kB PNG image 23.5 kB PNG image 24.1 kD PNG image 24.1 kD PNG image 24.1 kB PNG image 24.2 kB PNG image 53.8 kB PNG image	Complete file list	for recovery.img
Indeterminateo3 Indeterminateo1 Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o Icon_installing_o	661 bytes PNG image 673 bytes PNG image 773 bytes PNG image 23.4 kB PNG image 23.4 kB PNG image 23.4 kB PNG image 24.4 kD PNG image 24.4 kB PNG image 24.4 kB PNG image 53.8 kB PNG image	Complete file list	for recovery.img
indeterminate03 indeterminate01 icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o	061 bytes PNG image 067 bytes PNG image 073 bytes PNG image 23.6 kB PNG image 23.6 kB PNG image 23.6 kB PNG image 24.1 kD PNG image 24.1 kD PNG image 24.1 kB PNG image 24.2 kB PNG image 12.7 kB PNG image 12.7 kB PNG image	Complete file list	for recovery.img
Indeterminated3 Indeterminated7 Icon_installing_o Icon	661 bytes PNG image 673 bytes PNG image 773 bytes PNG image 23.9 kB PNG image 23.9 kB PNG image 23.4 kB PNG image 24.4 kB PNG image 24.4 kB PNG image 24.5 kB PNG image 12.7 kB PNG image 12.3 kB PNG image 10.3 kB PNG image	Complete file list	for recovery.img
Indeterminate3 Indeterminate40 Indeterminate01 Icon_installing_o Icon_in	661 bytes PNG image 673 bytes PNG image 733 bytes PNG image 23.8 kB PNG image 23.8 kB PNG image 23.8 kB PNG image 24.1 kD PNG image 24.4 kB PNG image 24.2 kB PNG image 24.2 kB PNG image 12.7 kB PNG image 10.3 kB PNG image 10.3 kB PNG image	Complete file list	for recovery.img
indeterminate3 indeterminate40 indeterminate01 icon_installing_o icon_installing_o icon_installing_o icon_installing_o icon_installing_o i con_installing_o i con_installi	661 Dykes PNG image 673 Dykes PNG image 773 Dykes PNG image 23.9 kB PNG image 23.9 kB PNG image 23.8 kB PNG image 24.1 kD PNG image 24.1 kB PNG image 24.1 kB PNG image 12.7 kB PNG image 19.3 kB PNG image 19.4 kB PNG image 1.4 kB pAnt ext document.	Complete file list	for recovery.img

5. Use the cpio tool to extract the content

1 item folder

```
$ cpio -i < ./Image</pre>
  3820 blocks
LCMXCMXCMX
             1 nanik nanik
                                    7 May 10 22:37 bbconfig
                                   7 May 10 22:37 blkid -> busybox
7 May 10 22:37 bunzip2 -> busybox
LFWXFWXFWX
             1 nanik nanik
             1 nanik nanik
lrwxrwxrwx
               nanik nanik
                                   8 May 10 22:37 busybox -> recovery
lrwxrwxrwx
                                   7 May 10 22:37 bzcat -> busybox
7 May 10 22:37 bzip2 -> busybox
               nanik nanik
lrwxrwxrwx
lrwxrwxrwx
             1 nanik nanik
                                   7 May 10 22:37 cal -> busybox
7 May 10 22:37 cat -> busybox
LFWXFWXFWX
             1 nanik nanik
             1 nanik nanik
lrwxrwxrwx
                                    7 May 10 22:37 catv -> busybox
               nanik nanik
FWXFWXFWX
               nanik nanik
                                    7 May 10 22:37 chattr -> busybox
lrwxrwxrwx
               nanik nanik
lrwxrwxrwx
                                    7 May 10 22:37 chgrp -> busybox
lrwxrwxrwx
             1 nanik nanik
                                    7 May 10 22:37 chmod -> busybox
              1 nanik nanik
                                     7 May 10 22:37 realpath -> busybox
8 May 10 22:37 reboot -> recovery
LFWXFWXFWX
              1 nanik nanik
lrwxrwxrwx
                nanik nanik 840564 May 10 22:37 recovery
- FWXF - X - - -
                                     7 May 10 22:37 renice -> busybox
                nanik nanik
lrwxrwxrwx
                                       May 10 22:37 reset -> busybox
                 nanik nanik
lrwxrwxrwx
              1
                                        May 10 22:37 resize -> busybox
                nanik nanik
lrwxrwxrwx
```

Busybox symlinks

The /sbin directory contains lots of

tools, most of which are a symlink to recovery (via busybox symlink) as shown in the screenshot.

system.img

The most important image file is system.img, which contains the heart and soul of Android, and is the largest in terms of file size.

1. Use the simg2img tool to extract the image file

```
$ simg2img system.img system.raw
computed crc32 of 0x9a5d4d54, expected 0x0000000
```

2. Using the file tool you will see the type of the extracted file.

```
$ file system.raw
./system.raw: Linux rev 1.0 ext4 filesystem data,
UUID=57f8f4bc-abf4-655f-bf67-946fc0f9f25b (extents)
(large files)
```

3. Create a directory to mount the extracted file and mount the file to the directory

```
$ mkdir sys
$ sudo mount -t ext4 ./system.raw ./sys
```

4. You will see a lot of folders inside the sys/ directory.

If you look each of the folders, you will the application, configuration and executables files that are used to run Android.

▶	арр	59 items	folder
▶	bin	178 items	folder
▶	etc	22 items	folder
▶	fonts	39 items	folder
►	framework	24 items	folder
►	lib	193 items	folder
▶	media	1 item	folder
▶	pseudorec	3 items	folder
▶	tts	1 item	folder
▶	usr	5 items	folder
▶	vendor	1 item	folder
▶	xbin	259 items	folder
	build.prop	1.8 kB	plain text documen

Complete file list for system.img

userdata.img

This particular image file resides in its own partition, and will be used as the /data partition. The image file does not contain many files, and sometimes, in different devices, it is just an empty image file. Since this is used for data, it does not matter what it contains, since Android does not use it, and is intended only for storage purposes.

The following steps shows you how to extract the userdata.img:

1. Use the simg2img tool to extract the image file

```
$ simg2img userdata.img userdata.raw
computed crc32 of 0x20aec0ac, expected 0x0000000
```

2. Using the file tool, you can see the type of the extracted file.

```
$ file ./userdata.raw
./userdata.raw: Linux rev 1.0 ext4 filesystem data,
```

UUID=57f8f4bc-abf4-655f-bf67-946fc0f9f25b (extents)
(large files)

3. Create a directory to mount the extracted file and mount the file to the directory

```
$ mkdir data
$ sudo mount -t ext4 ./userdata.raw ./data
```

4. You will see that there are only 2 folders inside the /data directory

(these 2 folders are not used in Android)

Packing

In the previous section, we unpacked the image file, so after making changes to the image files, you will want to pack them up again and use them to copy it into your sdcard or eMMC storage.

boot.img and recovery.img part 2

After making the changes inside the directory, you need to pack it up and convert it to a .gz file, using the following command.

\$ find . | cpio -o -H newc | gzip > ../newramdisk. cpio.gz

You have to run the above command inside of the extracted directory after you unpack it, using the cpio command as shown in the previous step.

A ../newramdisk.cpio.gz file will be created, and this file will be packed together using another command:

\$ mkbooting --kernel kernel.gz --ramdisk newramdisk. cpio.gz -o new_boot.img

The kernel.gz comes from the original boot.img, while the newramdisk.cpio.gz contains our changes. On completiong of the mkbooting command, you will have a new boot image called new_boot.img.

system.img part 2

After you make changes to the files inside the extracted system.img directory, you can pack it using the following command:

```
sudo make_ext4fs -s -l 512M -a system ./system_new.
img ./system
```

The last parameter ./ system is the directory that contains the extracted image files.

RESIZING ANDROID PARTITIONS MAKE FULL USE OF YOUR LARGE SD CARD OR EMMC

by Venkat Bommakanti

arge capacity installation media such eMMC or SD cards have a lot of unallocated space available after initial installation of a Hardkernel or community image. The partitions are kept small during the image creation process so that the resulting file is more portable, and can still be installed on smaller storage devices of 4GB or 8GB size. With a 16GB, 32GB or 64GB device, the partitions can be expanded in order to take advantage of the unused space, making it available for system use.

This article presents the steps necessary to properly resize your Android image's FAT32 partition, usually referred to as /mnt/sdcard. If the partitions are not resized properly, the resolution will be locked to 720p because the boot.scr file will not be found by the bootloader. The procedure essentially re-assigns most of the unallocated space to the Android partition while still allowing 1080p resolution - it's as simple as that!

Requirements

I. An ODROID U3 or XU board, with an appropriate power adapter

2. Any MicroSD or an eMMC card of 8+ GB capacity, along with an SD Card reader/writer, containing a freshly copied, unbooted Android image. To download the latest supported Android version relevant to your platform, check the following link, using http://bit.ly/lk80lbP. 3. A (native) Linux host with GParted (to graphically manage disk partitions) installed. If you have access to only an MS Windows system, you may use one of two options:

A GParted Live CD for x86 systems (http://bit.ly/lhGJVvw), or

An Ubuntu 14.04+ Desktop virtual machine instance hosted on the X86 system with Gparted installed. See http://bit.ly/lr20P48 and http://bit. ly/lnlDa5d) for details.

GParted for Linux is preferred over tools such as EaseUS Partition Master for Windows, since Gparted detects Android ext3/ext4 partitions properly without problems, but newer versions of EaseUS may not read ext3/ext4 partitions at all. So, it is advisable for MS Windows users to run GParted via USB stick or Virtual Machine, as mentioned above.

Overview

The first two steps involve moving two partitions to the end of the storage device (while still retaining their respective sizes), and

The third step is to combine most of the unallocated space to the already existing Android partition.

The first screenshot in this article shows the results of following the main

steps on a 32GB card. There are quite a few sub-steps, so make sure to carefully follow the figures below. Cards of a capacity greater than 64GB have not yet been tested, but should work properly if the extra room is created as a second FAT32 partition.

The reason for separate instructions when using a 64GB card, is that FAT32 partitions are limited to 32GB in Android, although more recent Android images allow for larger partition sizes. Even though most of the steps are shared between the procedure for 32GB (or less) and 64GB cards, we did this unified chart that can help you for either case.

In addition to the chart, there is also a pictorial guide for resizing the partitions in Gparted on pages 28 and 29.

If you have completed the steps correctly, the boot.scr file should be recognized properly and allow 1080p resolution. It's advisable to perform this operation on a new copy of Android to avoid any potential data loss. Always make a backup first!

⊗⊜ @ /de	ev/sdb - GParte	d					😣 🗐 🗐 /dev/sdt	o - GParted			-		
Parted Ed	lit View Devic	e Partition	Help		/dev/sdb (80.08 GiB) 🗘	GParted Edit Vi	ew Device Pa	rtition Hel V	p		/dev/sdb (3	0.08 GiB)
				unallocated 24.75 GiB					1	New Partition #1 26.76 GiB			
vartition unallocated /dev/sdb2 /dev/sdb1	File System	Label	Size 64.00 MiB 1.00 GiB 2.02 GiB	Used 	Unused 	Flags	Partition unallocated /dev/sdb2	File System unallocated ext4	Label	Size 64.00 MiB 1.00 GiB 26 76 GiB	Used 338.07 MiB	Unused 685.93 MiB	Flags
/dev/sdb3 /dev/sdb4 unallocated	ext4 ext4 unallocated		2.12 GIB 128.00 MIB 24.75 GIB	968.68 MiB 6.20 MiB	1.18 GIB 121.80 MIB		/dev/sdb3 /dev/sdb4	ext4	Android	2.12 GIB 128.00 MIB	808.68 MiB 6.20 MiB	1.34 GiB 121.80 MiB	
												8	
							👍 Create Primary	Partition #1 (fat.	32, 26.76 GiB	8) on /dev/sdb			
operations (pending					-	1 operation pendin	g					
Before	resizing th	e Android	partitions							After	resizing the	Android part	titions
For then use	a n eMMC o eMMC o e steps 6a	f or SDC r SD ca or 6b, tl	ard less thar urds less tha hen finish w	n 64GB n 64GB, fo rith steps 8-	ollow steps 10.	1-6,	Resizing The because	g a 64GB e procedur FAT32 _I	eMMC of the for reporting the formation of the formation	or SDCard esizing a 64 ns are limit	GB card is ed to 32G	s slightly di B in size.	iffere
First, hese st prepar e MMC card re	follow teps to re your or SD for its ssizing.	1. I 2. (FAT32 created 3. (nsert the ca Create a te 2 partition o 1 backup di Using GPar - unmour	ard or mod mporary d of the card rectory in t ted, perfor at the FAT.	ule into a l irectory of l. Backup the host. m the follo 32 partition	Linux h n the h the file owing o n,	nost using an nost's local es from the operations o	n approp drive, to card's FA on the FA	riate U receiv AT32 p T32 pa	SB adapter e the files artition to artition of	/reader. from the the newly the card:		
After ste	r doing ps 1-5,	4. \	delete th - select [A] - While still in	ne FAT32 _[Apply] to m 1 GParted,	partition, and the charter of the charter of the perform the charter of the chart	nd lastl ange p nese or	ly, ermanent. perations:						
mov	e on to 🛽				1	1							

unmount the last partition of the card,
move this last partition to the end of the card, and lastly,

- select [Apply] to make the change permanent.
- **5.** Then perform these final move operations:
 - unmount the 2nd-last partition

depending

on your eMMC

or SD card's capacity.

 6a. (<64GB) Move this 2nd-last partition towards the end, just before the previously moved (last) partition. The last two partitions should touch each other, with no empty (unallocated) space in between them. - select [Apply] to make the change permanent. 7a. (<64GB) When completed, the layout will look like so: 	 6b. (64GB) Move this 2nd-last partition roughly towards the midpoint of the unused space, such that: there is ~28GB space free to the left of the partition, and there is ~26GB space free to the right. select [Apply] to make the change permanent. 7b. (64GB) Edit the 2nd-last partition: regize it to use the ~26GB available space to its right.
 space in between them. select [Apply] to make the change permanent. 7a. (<64GB) When completed, the layout will look like so: a single untouched partition at the start of the card, 	 there is ~26GB space free to the right. select [Apply] to make the change permanent. 7b. (64GB) Edit the 2nd-last partition: resize it to use the ~26GB available space to its right,
 a lot of blank space, and then, two partitions all the way at the end. 	and lastly, - select [Apply] to make the change permanent.

8. Format the unallocated (available) space as FAT32 and select [Apply].

9. Copy the backup files from the local temporary backup directory (created in the second step) to the new FAT32 partition.

10. Reinsert the card into the Odroid device, boot and enjoy Android on your ODROID!

QUICK PICTORIAL GUIDE FOR RESIZING AN ANDROID SD OR EMMC

by Venkat Bommakanti

😣 🗐 🗐 /de	v/sdb - GParted					
GParted Edi	t View Device	Partition	Help			
🔓 🛇 🕒	4 G G	61			/dev/sdb	(30.08 GiB) 💲
				unallocated 24.75 GiB		
Partition	File System	Label	Size	Used	Unused	Flags
unallocated	unallocated		64.00 MiB			
/dev/sdb2	ext4		1.00 GiB	338.07 MiB	685.93 MiB	
/dev/sdb1	fat32	Android	2.02 GiB	493.07 MiB	1.54 GiB	
/dev/sdb3	ext4		2.12 GiB	968.68 MiB	1.18 GiB	
/dev/sdb4	ext4		128.00 MiB	6.20 MiB	121.80 MiB	
unallocated	unallocated		24.75 GiB			

Initial partition layout

. •		6 6 2				/dev/sdb (30.08 GiB)
		New	Insert	1	unallocated 24.75 GiB		
Partition	File Syst	Derize Moun	Detete		Used	Unused	Flags
unallocated	unallc	Resize/Move	· · · · ·	DO MIB	-		
/dev/sdb2	ext4	Сору	Ctrl+C	00 GiB	338.07 MiB	685.93 MiB	
/dev/sdb1	fat32	Paste		02 GiB	493.07 MiB	1.54 GiB	
/dev/sdb3	ext4	Eormat to	•	12 GiB	968.68 MiB	1.18 GiB	
/dev/sdb4	ext4	Mount		DO MIB	6.20 MiB	121.80 MiB	
unallocated	unallo			75 GiB			
		Manage Flags					
		Check					
		Label					
		New UUID					
		Information					

Move the selected I28 MB partition



Initial parameters of 128 MB partition to be moved

Resi	æ/Move /dev/sdb4			
				•
	Minimum size: 7 MiB Maxim	um size: 2	5467	мів
	Free space preceding (MiB):	25339	\$	
	New size (MiB):	128	\$	
	Free space following (MiB):	0	-	
	Align to:	MiB	-	
		Cancel		Resize/Move

New parameters to move 128 MB partition

S 🛇 🚽		1			/dev/sdb (30.08 GiB)
	New	Insert	u 24	nallocated 4.75 GiB		
Partition Fil	Delete	Delete	Size	Used	Unused	Flags
unallocated	<u>R</u> esize/Move	D	64.00 MiB	-	-	
/dev/sdb2	⊆ору	Ctrl+C	1.00 GiB	338.07 MiB	685.93 MiB	
/dev/sdb1	Paste		2.02 GiB	493.07 MiB	1.54 GiB	
/dev/sdb3	Format to		2.12 GiB	968.68 MiB	1.18 GiB	
unallocated 🔳			24.75 GiB	-	-	
/dev/sdb4	Mount		128.00 MiB	6.20 MiB	121.80 MiB	
	M <u>a</u> nage Flags C <u>h</u> eck Label New UU <u>I</u> D					
Move /dev/sdb	Information					

Move the selected 2.12 GB partition

8 Res	ize/Move /dev/sdb3			
6	Minimum size: 809 MiB Maxin	mum size:	2753	85 MiB
	Free space preceding (MiB):	þ	-	
	New size (MiB):	2176	\$	
	Free space following (MiB):	25359	-	
	Align to:	МіВ	1	
		Cancel		Resize/Move

Initial parameters of 2.12 GB partition to be moved

🛞 Resize/Move /dev/sdb3		
Minimum size: 809 MiB Maxi	mum size:	27535 MiB
Free space preceding (MiB):	25359	•
New size (MiB):	2176	\$
Free space following (MiB):	0	-
Align to:	MiB	-
	Cancel	Resize/Move
New parameters to move 2.12 GE	<mark>3 partit</mark> i	ion

ODROID BASICS

+

+

Add D

Flags

Close



HOW TO FEED YOUR CAT OVER THE INTERNET A GUIDE FOR ATTACHING STEP MOTORS TO THE ODROID-U3

by Marian Mihailescu

n the previous automation article, we showed how to attach sensors to the ODROID-XU: a temperature sensor and a motion detector to monitor a cat's movement habits. In this article we will go one step further, and learn how to control a step motor using the ODROID. And what better application for a motor than building an Internetcontrolled automatic cat feeder!

We will start with the Cat Mate C50 feeder, shown in Figure 1.



This particular feeder has a batterypowered clock mechanism that activates every 24 hours, releasing the red lever to the next position, which results in the white hood moving to the next meal compartment. It is also very easy to hack - you can just remove the clock mechanism that rotates the lever, and replace it with a motor, as shown in Figure 2 and Figure 3.

For this project, I will be using the ODROID-U3, which has several advantages over the ODROID-XU for this project: it's more affordable, requires less power, does not have a fan, and has the optional **IO-SHIELD** accessory, which provides extra GPIO



We interrupt this article to announce the return of the cat memes!

ports. The cat feeder will be controlled using a 5V stepper motor (model 28BYJ-48), which can be purchased from Ebay together with the driver board ULN2003 for less than \$3 USD. The stepper motor is ideal in this case, since it can rotate a fixed number of steps, allowing us precise



Figure 2a - Clock mechanism removed with lever gear visible

control of the cat feeder. In particular, the 28BYJ-48 motor has 32 steps per revolution, with a gear reduction of 1/64, which means that it does 2048 steps in one full revolution (360 degrees). Most motors usually need more current than development boards can support, and thus are usually connected to an external power



Figure 2b - Front access to the lever gear



Figure 3a - 28BYJ-48 5V stepper motor with ULN2003 Driver Board

Figure 3b - Motor fits right in the lever gear



supply or a battery pack; however this particular motor can get all the power it needs from the ODROID.

The role of the motor driver board is to allow the small current output from the ODROID to control a bigger current needed for the stepper motor. The driver board is controlled by four inputs that determine how the coil is magnetized in order to rotate the motor magnet core. There are also the 5V power and ground inputs, for a total of six connections to the ODROID.

Since the motor requires 5V logic, we need a level shifter to work with the 1.8V outputs of the ODROID. Similar to the previous automation article, where is explained in more detail, we will be using the four port Freetronics Logic Level Converter. Since the motor driver requires four GPIO pins for control and the ODROID U3 only has three GPIOs (GPIO199, GPIO200 and GPIO204), the ODROID IO-SHIELD is required. However, in the absence of the IO-SHIELD, it is possible to control the driver board with only three GPIO by using a shift register.

The shift register gets all required GPIO outputs in a serial way, and then outputs them to the stepper motor in parallel. You need to connect the SERIAL IN input to the GPIO that sends the motor control logic. CLOCK is used to make the shift register load the logical value from SERIAL IN on a low-high transition, while shifting the other stored values. LATCH enables the output of the shift register on a low-high transition. OUTPUT ENABLE



Even though a space hyperdrive would be far more useful, we dutifully succumb to the will of our feline masters by combining both robotics and Linux to automatically feed them, in case we suddenly "disappear".

is used to disable the outputs - since we want them enabled all the time, we can connect it to GND, and RESET is used to clear the shift register. We can connect OUTPUT ENABLE to 5V (VCC) and set up all the outputs each time, such that RESET is not required. This particular shift register model has eight outputs, so we will load eight values each time, even if we only need four.

The motor driver is connected to the motor in the feeder by a 2m cable, which allows the board to be placed safely away from the cat. To control the feeder, the bash script shown below is executed by clicking a button in a webpage. A webcam was also



Figure 4 - ODROID Internet-controlled cat feeder schematics

TECHNICAL ARTICLE



Figure 5a - Board with all components attached

added to the ODROID to provide visual feedback. Using Motion webcam server, a live stream is embedded in the webpage with the cat feeder controls. With a slightly more complex version of the script and the Linux cron job scheduler, it is possible to activate the feeder not only on-demand, but also automatically, at pre-defined periods of time. Your cat has never been happier!

```
#!/bin/bash
### enable ODROID U3 GPIO
echo 199 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio199/direction
echo 0
        > /sys/class/gpio/gpio199/value
echo 200 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio200/direction
echo 0 > /sys/class/gpio/gpio200/value
echo 204 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio204/direction
       > /sys/class/gpio/gpio204/value
echo 0
### setup motor sequence
declare -a SEQ0=(0 0 0 0 1 0 0 0)
declare -a SEQ1=(0 0 0 0 0 1 0 0)
declare -a SEQ2=(0 0 0 0 0 0 1 0)
declare -a SEQ3=(0 0 0 0 0 0 0 1)
### define level shifter GPIO
IN=/sys/class/gpio/gpio204/value
LATCH=/sys/class/gpio/gpio200/value
CLK=/sys/class/gpio/gpio199/value
### control motor
STEP_COUNTER=0
```

<section-header><section-header><section-header><section-header>

Figure 5b - Webpage to control the feeder over the internet

```
for STEP in {1..4096}
do
        echo 0 > $LATCH
        for PIN_COUNTER in {0..7}
        do
               PIN_VALUE="SEQ$STEP_COUNTER[$PIN_COUN-
TER]"
                echo ${!PIN VALUE} > $IN
                echo 1 > SCLK
                echo 0 > $CLK
        done
        STEP_COUNTER=$[$STEP_COUNTER + 1]
        if [ "$STEP_COUNTER" = "4" ]
        then
                STEP_COUNTER=0
        fi
        echo 1 > $LATCH
        echo 0 > $LATCH
        sleep 0.00000001
done
### after feeder activated, reset motor inputs
for RESET in 0 0 0 0 0 0 0 0
do
        echo $RESET > $IN
        echo 1 > CLK
        echo 0 > $CLK
done
echo 1 > $LATCH
echo 0 > $LATCH
echo "[$(date)] : Feeder activated successfully...
meow!"
```

MAKE A CUSTOM LEGO CASE FOR YOUR U3

DON'T HAVE YOUR OWN 3D PRINTER YET? ASSEMBLE THIS GREAT CASE FROM YOUR LEGO BUCKET!

by @Zylophone



HOW TO ENABLE MULTI-CHANNEL AUDIO OUTPUT WITH XBMC USING THE USB-S/PDIF PERIPHERAL

ODROID-U3

ODROID

AUDIO-LINK

Audio Receiver

Audio Flow Diagram

TO DELIVER DIGITAL 5.1 SURROUND SOUND

by Justin Lee

e have frequently received requests on the forums for instructions on how to enable 5-channel audio passthrough with ODROID computers. However, we couldn't find out how to implement the pass-through over the HDMI audio channel with the native hardware, even though we tried many different ways. So, we decided to develop pass-through functionality in another way, using a USB to S/PDIF interface.

Hardware setup

To allow HDMI passthrough, we created a USB-S/PDIF adapter board using the CM108AH chipset. We called it the USB-S/PDIF, which stands for AUDIO LINK. The price is only \$16 and it comes with a USB cable and plastic shell.

The Cmedia CM108AH is a highly integrated single chip USB audio solution. All essential S/PDIF optical TOSLINK





What does this setup need? An ODROID powering it, of course!

as well as Coaxial outputs, including PLL, regulator, and USB transceiver could be implemented on a small PCB.

Software setup

The passthrough function in XBMC 12 Frodo doesn't work with the ODROID, but fortunately, XBMC 13 Gotham is capable of enabling HDMI passthrough output. For best results, use the latest Ubuntu or Debian OS image which has the XBMC 13 preinstalled.

We tested our hardware with these 2 prebuilt OS im-

	Volume Control		- • ×		Volume Control	
ayback Record	ting Output Devices Input Devices	Configuration		Playback	Recording Output Devices Input Devices Configuration	
CM108 Aud	io Controller Digital Stereo (IEC958)			🐻 Built	in Audio	
Port: Digital	Output (S/PDIF)		•	Profile:	off	*
Front Left	-	• · · · · ·	100% (0.00d8)	E CM1	08 Audio Controller	
Front Right	1	•	100% (0.00dB)	Profile:	Digital Stereo (IEC958) Output	•
DTS Latency offset	MPEG	Cons	- + ms			

PulseAudio Configuration Settings

TECHNICAL ARTICLE





ages, either of which may be used to achieve the necessary software configuration for 5-channel surround:

```
http://forum.odroid.com/view-
topic.php?f=29&t=4823
```

http://forum.odroid.com/viewtopic.php?f=79&t=4626

To begin installation, plug in the USB-S/PDIF and boot your board. In order to activate Passthrough Codecs, first setup PulseAudio Volume Control according to the screenshots. Select Digital Output (S/PDIF), then select AC3 and DTS in the Advanced option menu.

Finally, start up XBMC, where you can easily setup the passthrough function. Select the CM108 device in the Audio output menu, then enable the options called Enable passthrough, Dolby Digital AC3 capable receiver, and DTS.



Setup on XBMC

The receiver which we used for this tutorial is a very cheap entry level, which works well for development and testing.

To verify that the hardware and software are configured properly, playing a video with AC3 sound should display that Dolby Digital encoding is being







Our audio system displaying both Dolby Digital and DTS sound. 5.1 surround is awesome!

used by the amplifier, and playing one with DTS sound should indicate DTS encoding.

There are also several third-party USB sound cards that use S/PDIF output which should work as 5.1 passthrough devices as long as the Linux driver supports it. Please note that PulseAudio does not currently allow TrueHD or DTS-MA passthrough, which is a PulseAudio limitation, unrelated to the USB-S/PDIF hardware or XBMC software configuration.

The USB-S/PDIF is now available from the Hardkernel store for \$16 at http://bit.ly/lkIt8aC.

TRAVEL BACK IN TIME WITH TELNET DUST OFF THAT OLD 1200 BAUD MODEM

by Rob Roy

efore the World Wide Web and HTTP was invented, a protocol called Telnet was a popular way to communicate from one computer to another. By creating a primitive form of a web page that operated over Telnet, computer hackers built a large community of unique text-based server nodes running software called BBS (Bulletin Board System). This network included many special-interest communities that shared unique hobbies such as role playing, music and programming. To take a trip back in history to 1984, when the BBS network was in its golden age, type either of the following into a Terminal window:

telnet 1984.ws 23 telnet gcomm.com

Register and enjoy some of the most advanced ASCII graphics from the early days of network computing! More nostalgic BBS experiences such as multi-user dungeons (MUDs), chat programs and other epic systems can be found at http://bit.ly/lqnkmSq.



OS SPOTLIGHT: DREAM MACHINE AND WHISPER UBUNTU 13.04 AND 13.10 WITH 8 DESKTOP

ENVIRONMENTS, ALSO AVAILABLE IN TURBO

Dream Machine and Whisper with Kernel 3.8 for the U2/U3/X/X2 may be downloaded from http://bit.ly/IrhHymu

by Rob Roy, Editor-In-Chief

ream Machine, a community prebuilt OS image, offers nearly every flavor of Ubuntu desktop available for versions 13.04 and 13.10, including Lubuntu, LXDE, Kubuntu (KDE), Blackbox, Openbox, Matchbox, Unity and Xubuntu (XFCE). The desktop environment can be selected by using the gear icon in the upper corner of the Dream Machine login screen. Whisper comes with a single performanceoriented speed-optimized environment called Lightweight X11 Desktop Environment (LXDE), which is graphically very fast and offers a similar interface to Windows.

Because the software on Dream Machine includes everything on the Fully Loaded Ubuntu 12.11 image, please refer to the previous OS Spotlight column (May 2014) for details on the included packages. However, Dream Machine also

KDE Plasma Workspace features Open GLES2 GPU acceleration and has gorgeous effects! comes with XBMC, which is not included with Fully Loaded, since XBMC only works with ODROID Kernel 3.8. There have been several improvements to 1080p playback in XBMC since Dream Machine 13.10 was finalized, and the latest version of XBMC will be included in the upcoming Dream Machine 14.04 release.

KDE Plasma Workspace

Although every desktop environment on Dream Machine contains basically the same software, there are many differences in the look of each desktop as well as additional UI features. The default environment in the 13.10 version is KDE Plasma Workspace, which features beautiful hardware-accelerated fade and transition effects in its windowing environment. KDE comes with a driver for OpenGLES called kwin_gles, so that the animations remain smooth and don't stress the CPU. Because other Ubuntu desktops are (generally) written for OpenGL, which is not supported by the Mali GPU, rendering in those environments is done at the software level, but KDE offers very fast UI responsiveness.

KDE Plasma Workspace will be familiar to users of Windows operating systems, since it has a well organized Start Menu and uses a similar desktop layout, with a notification area in the bottom right. It has its own robust screen saver program, and is the only environment to include dozens of notification sounds. It also has a Recent Documents and Applications area, as well as many useful widgets such as clocks and weather. It's my favorite desktop, and is a great choice for office work or web browsing. For more information on the KDE desktop, or to contribute to the ongoing open-source project, visit http://www.kde.org/.

Blackbox adheres to a minimalist aesthetic.



OS SPOTLIGHT



LXDE (Lightweight XII Desktop Environment) balances performance, speed and features.

Blackbox

The simplest desktop environment on Dream Machine is called Blackbox, which displays only a task bar and a compact Applications menu, accessed by right-clicking the Desktop. It's meant as a very lightweight way to monitor system processes, or just to reduce CPU cycles for power-hungry applications. A typical scenario for using Blackbox is an embedded Linux system that runs a single custom application, such as a robotics controller. The design is visually minimalist, it runs applications quickly, and also has a small memory footprint.

LXDE

LXDE stands for Lightweight X11 Desktop Environment, and is best suited for those who wish to get the maximum performance from the ODROID while still retaining a robust interface. It looks and feels like Windows XP, and excludes graphic compositing and other visual ef-

Matchbox is designed for PDAs, but also looks sharp on an HD 1080p monitor.

fects in order to boost the environment's efficiency.

XBMC runs very well using LXDE, and is the best choice for those who wish to setup a Linux set-top box running XBMC, Transmission or Youtube. LXDE also offers a unique File Manager called PCManFM, which leverages the gio/gvfs libraries to connect to sftp, webdav, smb, and other Gnome Virtual File System-supported protocols.

Lubuntu

Lubuntu, which is based on LXDE, offers some visual improvements to the basic LXDE interface with larger, squared-off icons and a streamlined taskbar. It also features many new software packages, including the Lubuntu Software Center.

Matchbox

Matchbox is different from other environments in its use of the entire desktop as an Applications menu. Click the arrows in the upper left and right to page through the categories, and scroll down the desktop to see more applicaLubuntu is built on LXDE and offers a wide range of unique software applications.

tions. Matchbox is perfect for accessing programs and icons quickly without needing to constantly open a menu, since every application shortcut is displayed on a single page. Although Matchbox was designed for PDAs and tablets, it looks great on a 1080p HDMI monitor. Matchbox shows only one window at a time rather than using a multi-window layout.

Openbox

Openbox isn't actually a desktop environment, but a stand-alone Window Manager. It can be run by itself or within another desktop environment, and uses the minimum amount of resources necessary to launch programs and manage windows. The main focus of Openbox is speed, and is ideal for time-sensitive programs, high-traffic servers, and other applications where memory and processor use is important. Openbox is extremely sparse, has very few dependencies, and is delightfully compact.

Openbox comes with only a terminal and web browser, and is perfect for embedded systems.

in advand i bas	i - Kansule		
File Edit V	ew Bookmarks Settings Help		
Invenience 1 Invenience 1	661 6 Jan 3 2013 k2tgrep -> k2grep 661 5421 Jan 8 2013 kzgrep	â	
Anne and a second se	oot 1807 Jun 3 2013 sztess oot 2168 Jun 3 2013 szeare		
Invenience 1 Invenience 1	cot 22 Apr. 2 2013 sacc -> /etc/alternatives/yacc cot 5556 Pob. 7 2013 sletephe		
- PART-REAR 1	001 20112 Sep 5 2013 yelp 001 10120 Jan 16 2013 yes		
Department 1	eet DEEE Feb 7 2013 ywwsplittappe		
INTERACT I	001 9600 0:1 11 2013 200mp	g reddt: the front page of the laterant - Chronium	
-Network 1	oot 120052 Sep 25 2003 reitgeist-dareen	reddt: the fort page o x	
-TWEF-LEF-X	11 CHOID STP 25 2013 CHIEFSS HATANAN	🔶 🌵 😂 🗋 www.reddt.com	• f
AND ADDRESS OF A	oot 63056 apr 20 2013 zipclaak	MY BURKEDETS W HIGHT - ALL - RAVIDOM MEDLYDYTERESTING - PUTUROLOGY - 6576 - 1AMA - SPORTE - 8 20	DIOWERTHOUGHTS - NEWS - BAMDIG - DOCUMENTARIES - ART - MOVIES - DF - N
 Detectors 1 Detectors 2 	oot 2953 May 13 2013 zipgreg oot 116478 May 13 2013 zipinfe	Greddit we are take astronation to allow white another	want to com? login or register in seconds []
- Detroite 1	oot 56676 Apr 29 2013 zipeete oot 56676 Apr 29 2013 zipeete	VIII III IIII IIIIIIIIIIIIIIIIIIIIIIII	TTTTTTTTTTT
OWER-REAL	oot 22152 Sep 2 2013 zjsdecade oot 9732 Jul 10 2013 zlib-flate	reddit's stories are created by its users	
of we react out 1	ool 34456 Jun 20 2003 zseetim -s D	in the community, vote, and change the world.	trouted evenues
10	advad - bash	2 banding subreddits: A/DDRR (Ohaydditamath (Olyhophaads (Onawitotama (Olidomathat)	1 Il comments
		+ 📲 📲 The difference 18 years makes (Language)	Submit a new lok
		solution of a submitted 2 hours age by manage_reinbardt to Adploy the common to share	
		 And when I got my EVO they said it was an impractical ca 	M1 (anyor card) Submit a new text post
		104 and 104 consistent of house age by Tabland to (Warwy	deity relation which would
		+ Balance of power 1990 game over screen. Issues and	
		⇒ 2012 ↓ 105 converte share	help support rochte
	Terminal emulator	 FIFA sponsor Adidas joins Sony in voicing concerns about Qatar's w 	vinning bid for the
	Web browser	solveitind 7 bours age by BulliadVanlaw to (dyaddown)	
	Decision Games	tot converse there	
	ObConf Hitput	5 39773 Source of Neurol Detrice (ubasequee)	
	Reconfigure	 The second state The second state The second state 	20 holograms of the samer live onto fields across the
	Record	1 3545 world. (posta on)	ab holograms of the games live onto helds ad oss the
	U.S.	 Weinerstein and State a	
		+ IS Great Single Location Movies (maximum)	
		↓ Setterstand Theory age by Taffy711 to Atmosphere ↓ 1488 conversely share	
		+ Water dripping from the cliff has eroded sand around thes	se shells, forming small sand towers with shell roofs.
		 A control of any procession of any procession of a second procesion of a second procession of a second procession of a secon	
		23 comments share	

onurp	on an m	10000						
Leafpad Simple text edtor	Leave	Liferea Download and vi	Lights Off	Log Out Log out of the Xtc	USE the comman	Mail Reader Read your email	Add or remove a	Associate applic
Mines Clear hidden min	Monitor Set Change screen r	Mouse and	Simple Test Editor	Mouse & To Set your mouse	Music Player Play your music fi	Network Network settings	Network Co Hanage and cha	Network To. View information
Nibbles Oude a vorm ar	Motes Ideal for your put	Notifications Customize how n	Onboard Itexible onscreen	Onboard Se Change Onboard	Online Acco Manage online ac	Openbox Leg in using the	Openbox Co Tweak settings fs	OpenJDK Java 7 0
Orage Calen Desktap calendar	Orage Glob Shew clocks from	Settings for the X	Panel Customize the p	Panel Mana Marage panel co	PDF Mod Remove, extract	Personal Fil Preferences for s	Pidgin Inter Onst over M. Su	Power manager
Power Man Settings for the X	Preferred A	Preferred A	Change proters	Configure printers	PulseAudio View and modify	PulseAudio	Qt 4 Settings	Qt Configur
Quadrapassel	Razor Appe Configure appear	ZZ Razor Autos	Razor Confi	Razor Deskt	Razor Freed	Razor Mous	Razor Sessi Configure Razor	Razor Short
Region & La Change your regi	Cannact to rame	Removable configure manage.	Ristretto Im	Robots	Bun Progra	Rygel Prefe	Samba Create, modily, a	Screenshot
Scribus Fage Layout and	Sensor Vie	Session and	Settings Edl Graphical comeg	Settings Ma Graphical Setting	Shotwell Criganize your ph	Simple Scan Scan Documents	Software U Show and instal	Software & Configure the so
Software &	Change sound ve	Sound Change sound vo	Sudoku Seet your logic sk	Gear the screen	Synaptic Pa	Synaptic Pa	System Loa A system load m	System Mo.
System Set	Deat the odds in	Task Manager	Terminal Em Use the comman	Complete the pu	Text Entry Charge your key	Centigure GTE th	Thunar File Drowse the Risty	Cond and received
Change system 1	Tomboy Not Taka rotes, link L.	Transmission Downlead and sh	Ubuntu Soft Lets you choose f	Universal Ac	User Accou	Users and G	UXTerm standard terminal a	Wacom Table
Web Browser Browse the web	Web Browser Browse the web	Window Ma Configure window	Window Ma Fine tune window	Wireless Mo BBBB Fanel based wrel	Wingul Simple Gift gui to th	Workspaces configure layout,	Xaos - Fract Hactal Generator	Xarchiver
XBMC MedL.	XChat IRC	Xfburn	Xfce Terminal	XTerm		Þ		14:4

OS SPOTLIGHT



Razor is written entirely in the advanced QT programming language.

Razor

The Razor-qt desktop is written entirely in the QT programming language, and is one of the more recent desktop environments to be included in Ubuntu 13.10. While still a new project, Razorqt already contains many key components, such as a panel, desktop, application launcher, settings center, and user session. Built for versatility and reuse, Razor-qt runs in conjunction with any number of window managers, although most Razor developers prefer to use Openbox. It's possible to use Razor with any modern window manager, from gwwm2 to kwin, including running KDE without the Plasma Desktop.



Xubuntu is a favorite of Ubuntu enthusiasts and offers many user-friendly features.

XFCE and Xubuntu

Xubuntu is based on the XFCE desktop environment, with the addition of many useful bundled packages which are meant to replace many Gnome-based software applications. It has a useful shortcut bar in the bottom area that appears on mouse hover, and an upper left start menu like Gnome. XFCE is a favorite of many Ubuntu users because of its nice balance between speed and functionality, as well as its convenient organization of applications into categories. The Search feature, represented by a magnifying glass in the shortcut bar, is similar to the Unity search bar, and can be used to find any program easily.



XBMC is the go-to program for all media playback, and can launch programs too.

XBMC

XBox Media Center (XBMC) can run as a standalone application, but can also be enjoyed as its own desktop environment by installing the "xbmc-standalone" package from Synaptic Package Manager. It is an all-in-one solution for audio and video playback, supporting Samba server connections, 5.1 DTS playback, third-party add-ons, and beautiful audio visualizations. It can also be used to launch Linux programs, so that software packages can be accessed within the need to go back to an Ubuntu desktop. For more information on using XBMC, please read the Quick-Start Guide at http://bit.ly/lkTXKXI.

Whisper (LXDE)

Whisper is a single-user version of Dream Machine that offers one desktop environment (LXDE) and boots straight to the desktop very quickly by setting the auto-login variables in /etc/lightdm/lightdm.conf. It is designed for performance and speed, and contains the same major software packages as Dream Machine. Whis-



per is one of the most popular images available on the forums, and provides a familiar Windows XP-style interface which is ideal for programming, video playback, CPU-intensive applications, or just for lightweight use such as web browsing and document editing.

Although Dream Machine is not yet available for the XU series, Whisper has been successfully ported to the

ported to the XU, and offers GXine instead of XBMC for video playback, since the XU version of XBMC does not run within X11. Open any media file from the File Explorer to automatically launch GXine, and double-click the playback window to enable full-screen viewing.

Now that that the Hardkernel 14.04 images have been released, an updated version of Dream Machine and Whisper will be released to the forums in the next few months. The recent improvements to XBMC that enable smooth 1080p playback will also be included in the X and U versions. Hardkernel is constantly improving its 3.8 kernel branch, and has recently replaced the kernel-update.sh script with the new odroid-utility package, which will also be added to the new releases.

Whisper, as fast as your ODROID can go!

YOU'VE GOT MAIL... **OR SHOULD!**

SUBSCRIBE TO THE HARDKERNEL EMAIL LIST TO KEEP UP WITH THE LATEST **NEWS IN THE WORLD OF ODROID!**

by Bruno Doiche

The latest software release for ODROIDs

Ubunut 14.04 release for ODROID-U3/U2/X/X2 boards

* Canonical Lubuntu 14.04 LTS base

- Mali GPU r4p0 drivers + armsoc driver + patched Xorg server
- Kernel 3.8.13.23 with improved 2D/3D hardware acceleration drivers
- * All in one ODROID-Utility for HDMI configuration, Kernel-update, root partition expander and much more tools for maintenance
- * XBMC 13.1 Gotham with Exynos-4412 MFC/FIMC hardware acceleration for 1080p videos
- * Hardware accelerated Gstreamer plugins.
- * Details : http://forum.odroid.com/viewtopic.php?f=29&t=5234

Android KitKat 4.4.3 for ODROID-U3/U2/X/X2 boards

- The latest Android OS KitKat 4.4.3 has been ported to ODROID.
- * Source code and update images are available : http://com.odroid.com/sigong/nf_file_board/nfile_board.php

Find out the most latest ARM technology from ODROID Magazine, a free monthly PDF e-zine!

POWERED ROBO

- MAY ISSUE
- Android Booting Process: Understand the Innards of How Your ODROID Boots Up to Android
- Pipe Viewer: Get More Interactive With Your Data Progress Tools
- The Force is Strong with Traceroute
- How to Compile Doom on Your ODROID: Play this Timeless Classic Custom Compiled for Your Machine
- Recompile the Mali Video Drivers: Fixing the "Blank Screen" and "Slow Windows" Issues when Upgrading to Ubuntu 14.04
 2 Systems, 1 ODROID, Pure Fun! How to Make a Dual Boot System with Android and Ubuntu
- Getting Started with Your ODROID: How to Copy an Image File to an SD Card or eMMC
- Get Yourself a Little More Personality on Your Sudo
 - Find Your Larger Files on Your Directory
- Split a Huge File
- On the Thermal Behavior of ODROIDs: The Performance Difference Between the XU and U3 in Greater Detail
- Indiegogo Campaign Promises ODROID Compatibility With Stretch Goal
 - ODROID-SHOW: A Powerful Mini LCD Screen for the U3
 - ODROID-UPS Kit: The Zero-Downtime Solution For the Rest Of Us
 - OS Spotlight: Fully Loaded Ubuntu 12.11 With Unity 2D Desktop Environment
 - Monitor Your Linux with Nmon
 - Build an ODROID-Powered Off-Road Unmanned Ground Vehicle: Part 1 Overview, Platform Assembly, and Power Distribution Meet an ODROIDian: Simone (@Sert00), A Long-Time ODROID Enthusiast and Helpful Computer Expert



- APRIL ISSUE
- Build Android on ODROID-U3: From Scratch to Smash, Take Total Control of Your Android System
- Shairport: Turn Your ODROID to an iTunes Airport Audio Station
- Portable Image Backup: Creating a Recovery File for Your Favorite Operating System - Rename Your Files from Uppercase to Lowercase in One Command Line
- Protected Yourself from Superuser Accidents
- Build Your Own Ubuntu From Scratch: Using Linaro's RootFS To Compile Linux Like a Pro!
- How to Install the Oracle Java Development Kit (JDK) Version 8: Save Time with Java's "Code Once, Run Anywhere" Architecture
- Using ODROIDs in High Performance Computing: What a Difference a Kernel Makes
- Android Gaming: Vector, Parkour Packed Action
- How to Setup a Minecraft Server: Creeeepers!
- Download Youtube Videos to Watch Offline
- Create a Papercraft Doll to Go Alongside Your Minecraft Server
- Learn Rebol: Writing More Useful Programs with Amazingly Small and Easy-To-Understand Code
- Be Heard With Ubercaster: A Real-Time Audio Broadcaster Hotspot - ODROID-U3 I2C Communication: Inter-Integrated Circuits for the Rest Of Us
- Heavy-Duty Portable Linux Tablet with LTE Router
- How I Built a Truck PC with my ODROID: Nevermind the Products on the Market, Get the Most Bang for Your Buck!
- Meet an ODROIDian: Marian Mihailescu One of Our Top Forum Contributors

ODROID MAGAZINE 39

a new OS release is about to happen? How about the contents of every ODROID Magazine issue in one list? Email us today - we won't bother you with mundane news, just the cool stuff! odroid@hardkernel.com

Want to know when

MEET AN ODROIDIAN RUPPI KIM, ONE OF THE FOUNDING MEMBERS OF HARDKERNEL

edited by Rob Roy

Please tell us a little about yourself.

I am the CTO of Hardkernel, as one of its founding members. I have been living in South Korea for 39 years and have worked in the hardware and software engineering industry for about 20 years. PCB artwork was my first job until I did my military service at 18 years old. After military discharge, I went into hardware and software development.

What do you work on at Hardkernel?

I am a hardware systems engineer.

I have been designing circuits related to the ODROID computers, and artwork the PCB to the main boards. In my extra time, I port the latest Linux kernel, along with purchasing, manufacturing, and everything related to it. I also manage the odroid.com server and Hardkernel GitHub repository.

How did you get started with computers?

I started in middle school in 1998. I remember the first time that a computer showed up in class for a pilot course. It was first grade (I think I was 13), and the computer was a DAEWOO IQ1000 (MSX). Before I discovered computers, I only knew computers by the brand names in the arcade game room.

I first learned a programming language that allowed drawing of shapes with characters and arithmetic called MSX BASIC. Here is the spec sheet for the IQ1000 that I used in school: http:// bit.ly/lplelKM.

After 3 years, I bought a \$1500 NEW-TEK 286 (AT) model as my first computer.

MEET AN ODROIDIAN



이제, 대우퍼스컴 아이큐 1000으로 세계 어느 MSX방식 컴퓨터의 소프트웨어도 쓰실 수 있읍니다.

ATTENDE MET SHARES
 THE SHAR

대우퍼스컴아이큐 1000

Our art editor was enthralled with this picture of Bakyeon Falls in winter, along with the chance to publish a photo of an old MSX!! This article couldn't get any better...

... Unless, of course, we get to include pics of cute babies! Welcome Kyung A and Kyung Eun!

It had an INTEL 80286 (Code Name: P2) microprocessor at a clock speed of 16MHz. Turbo was at a clock speed of 21MHz. It used a Trident 1MB SVGA graphic card with a Goldstar (LG Electronics) 14" monitor, and had 1MB of memory. I mainly learned macro assembly language and typing exercises with this computer.

1 Carl 1 - 10-20

What types of projects have you done with your ODROIDs?

In the future, my ODROIDs will be used for educational purposes with my new twin daughters who were born in the autumn of last year. Caring for twin girls has become my new hobby! Their names are Kyung A and Kyung Eun. When not with my children, I am very involved in Linux development and the new upcoming Linux kernel: http://bit.ly/SehCIb.