BuildRoot Media Player • Touch MIDI • Linux Gaming + Dancemania

ODROID Magazine

Year Two
Issue #20
Aug 2015

THE EVOLUTION OF COOL

The ODROID-XU4 combines the best of the U3 and XU3 into an awesome new device!

Build Android KitKat on the ODROID-C1

• Use wireless PS3 controllers on your ODROID

What we stand for.

We strive to symbolize the edge of 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.

HARDKERNEL

Pollin ELECTRONIC

Elektronik, Technik und Sonderposten zu Superpreisen!

Sie sind hier: Startseite > Suche

Suchergebnis Ihre Suche nach "odroid" ergab 18 Treffer

1 - 18 von 18 Ergebnissen

ODROID-U3 BACKUP BATTERY
RTC Lithium-Backupbatterie für den ODROID XU, XU+E, XU Lite und U3. Wird direkt auf den RTC-Port des ODROID gesteckt.
2,95 €*
Best.Nr. 810 319

ODROID-U3 Einplatinen-Computer, Cortex-A9 QuadCore, 2 GB
Kleiner als eine Scheckkarte aber riesig in der Leistung! Der ODROID-U3 verfügt über eine leistungsstarke SAMSUNG QuadCore-CPU, 2 GB Arbeitsspeicher und
69,95 €*
Best.Nr. 810 264

ODROID-U3 ODROID-UPS, USV, LiIon, 3 Ah
Unterbrechungsfreie Stromversorgung mit zwei LiIon-Akkus. Das Board besitzt die gleichen Außenmaße wie der ODROID-U3 Einplatinencomputer und kann direkt per
31,95 €*
Best.Nr. 810 328

ODROID-U3 eMMC Modul, 64 GB, mit Linux
Superschnelles NAND-Speichermodul für den ODROID-U3 Einplatinencomputer. Das Linux-Betriebssystem XUBUNTU ist fertig aufgespielt und kann direkt

ODROID-U3 USB-UART MODULE, Schnittstellenkarte
USB/UART System-Interface (IC: CP für die ODROID Einplatinencomput X2, U2, U3, XU, XU+E, XU Lite). I Entwicklung und Debugging.
15,95 €*
Best.Nr. 810 326

ODROID-U3 CASE Kunstst
Passgenaues Gehäuse mit Lüftungsschlitzen für den Maße (LxBxH): 87x52x29
4,95 €*
Best.Nr. 810 317

ODROID-U3 eMMC Android
Superschnelles N den ODROID-U3 Betriebssystem aufgespielt un
39,95 €*
Best.Nr. 810

ODROID-Linux
Supersc den OD Linux-berei
39
Bes

Ihre Vorteile
✓ Innovative Produkte
✓ Optimales Preis-/Leistungsverhältnis
✓ Kompetente Beratung
✓ Schnelle Lieferung

Noch Fragen?
Wir sind gerne auch persönlich für Sie da!
+49 (0) 8403 / 920-920

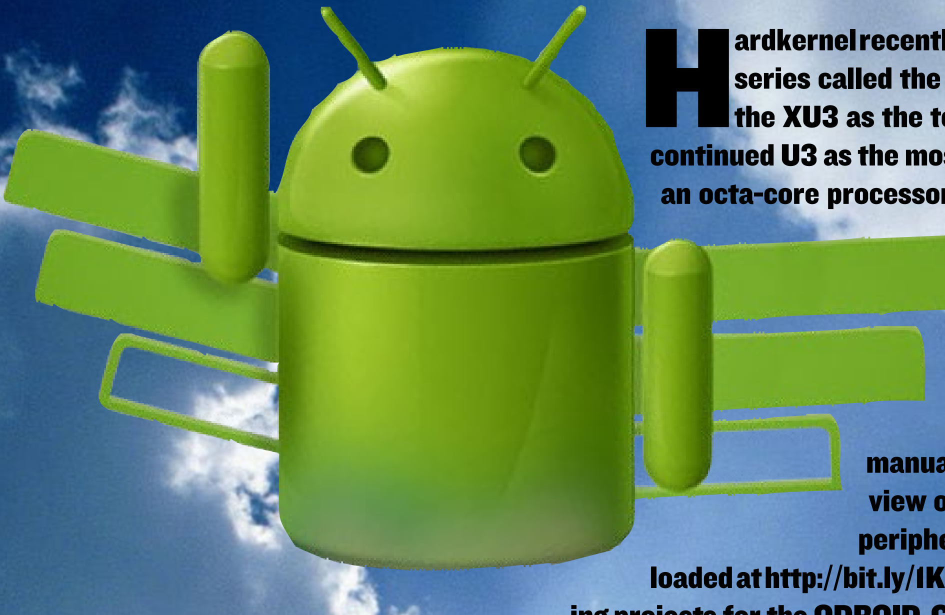
Sicher Einkaufen

We are now shipping the ODROID-U3 device 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://bit.ly/1tXPXwe



Hardkernel recently released the latest generation of the XU series called the ODROID-XU4, which not only succeeds the XU3 as the top-end model, but also replaces the discontinued U3 as the most versatile ODROID available. It boasts an octa-core processor which is compatible with all XU3 software, along with several improvements such as a smaller footprint and a more affordable price. The XU4 may be ordered from the Hardkernel store at http://bit.ly/1KhFr6d. The ODROID-C1 also had its official user manual released last month, offering an overview of the board with details about the many peripherals available for it, which may be downloaded at http://bit.ly/1K2NvMm. There have been many interesting projects for the ODROID-C1 since its introduction earlier this year, but one of the most unique applications is featured this month: a fruit-based MIDI piano! Venkat continues his technical series with a look at Grails, Tobias delights us with an introduction to a fantastic classic game called Dune, and Nanik shows off his method of building Android KitKat from scratch. We also bring out the fun side of ODROIDs with a review of Stepmania, a game that lets you dance your way to victory.

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, 431-815
Hardkernel manufactures the ODROID family of quad-core development boards and the world's first ARM big.LITTLE single board computer.
For information on submitting articles, contact odroidmagazine@gmail.com, or visit http://bit.ly/1ypImXs.
You can join the growing ODROID community with members from over 135 countries at http://forum.odroid.com.
Explore the new technologies offered by Hardkernel at http://www.hardkernel.com.

HARDKERNEL

HARDKERNEL'S EXCLUSIVE NORTH AMERICAN DISTRIBUTOR

ameriDroid
High-Performance Embedded Computers

Touchscreen quad-core computer for under $64! (ODROID-C1 and 3.2" Touchscreen)

SHOP NOW

USB GPS MODULE
$26.95

ODROID-C1
$36.95

ODROID-VU
$119.95

C1 3.2 INCH TOUCHSCREEN DISPLAY SHIELD
$26.95

All Hardkernel products in stock at AmeriDroid.com

ODROID Magazine

Rob Roy, Chief Editor

I'm a computer programmer living and working in San Francisco, CA, designing 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/1fsaXQs`.

Robert Cleere, Editor

I am a hardware and software designer currently living in Huntsville, Alabama. While semi-retired from a career in embedded systems design, including more than a decade working on the Space Shuttle program, I remain active with hardware and software product design work as well as dabbling in audio/video production and still artwork. My programming languages of choice are Java, C, and C++, and I have experience with a wide range of embedded Operating Systems. Currently, my primary projects are marine monitoring and control systems, environmental monitoring, and solar power. I am currently working with several ARM Cortex-class processors, but my ODROID-C1 is far and away the most powerful of the bunch!

Bruno Doiche, Senior Art Editor

Is having the most fun with his pair of new XU4 devices, and also received the best gift ever from his daughter for father's day. The only thing that he is missing is the inspiration for coming up with clever hostnames for his new machines.

Nicole Scott, Art Editor

I'm a Digital Strategist and Transmedia Producer specializing in online optimization and inbound marketing strategies, social media directing, and media production for print, web, video, and film. Managing multiple accounts with agencies and filmmakers, from Analytics and Adwords to video editing and DVD authoring. I own an ODROID-U3 which I use to run a sandbox web server, live in the California Bay Area, and enjoy hiking, camping and playing music. Visit my web page at http://www.nicolecscott.com.

James LeFevour, Art Editor

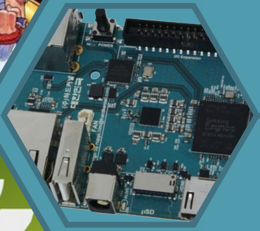
I am a Digital Media Specialist who is also enjoying freelance work in social network marketing and website administration. The more I learn about ODROID capabilities, the more excited I am to try new things I'm learning about. Being a transplant to San Diego from the Midwest, I am still quite enamored with many aspects that I think most West Coast people take for granted. I live with my lovely wife and our adorable pet rabbit; the latter keeps my books and computer equipment in constant peril, the former consoles me when said peril manifests.

Manuel Adamuz, Spanish Editor

I am 31 years old and live in Seville, Spain, and was born in Granada. I am married to a wonderful woman and have a child. A few years ago I worked as a computer technician and programmer, but my current job is related to quality management and information technology: ISO 9001, ISO 27001, and ISO 20000. I am passionate about computer science, especially microcomputers such as the ODROID and Raspberry Pi. I love experimenting with these computers. My wife says I'm crazy because I just think of ODROIDs! My other great hobby is mountain biking, and I occasionally participate in semi-professional competitions.

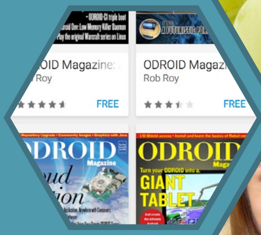
INDEX

SONY
SELECT
START

PUZZLE FIGHTER II
TURBO


KERNEL
USER MANUAL

STEP
MANIA


ODROID Magazine
Rob Roy
FREE
GIANT TABLET

PS3 WIRELESS CONTROLLERS

YOUR DRIVERS FOR WIRELESS FUN

by Tobias Schaaf

After 10 years on the market, if you don't have at least a pair of these, you will now certainly get them!

If you own a PlayStation 3 (PS3) controller and want to use it on your ODROID, here's an easy way to setup your controller to work wirelessly using a bluetooth adapter.

Requirements

- **USB Cable for your PS3 controller**
- **A supported bluetooth adapter connected to your ODROID**
- **sixad drivers from my repository**

Installing drivers

There are two methods of installing the drivers. The first is to add my repository to your system, which offers many games and useful software packages. As root, run the following commands:

```
# cd /etc/apt/sources.list.d/
# wget -O meveric-all-main.list http://bit.ly/1gu4vjj
# wget -O- http://oph.mdrjr.net/meveric/meveric.asc \
  | apt-key add -
# apt-get update
# apt-get install sixad
```

The other method is to install it manually:

```
# wget -O sixad_1.5.1.deb http://bit.ly/1Jrn9l6
# dpkg -i sixad*.deb
# apt-get install -f
```

Pair the controller

Connect your controller via USB cable for the initial configuration. As root, run the following command:

```
# sixpair
```

Next, unplug the USB cable, and type the following as root:

```
# sixad --start
```

Press the PS button on your PS3 controller, and after a short while, you should feel it vibrate and the status LED on your PS3 controller should light up, which means that it's working. The Sixad daemon will run automatically after the next reboot. Whenever you want to use the PS3 controller, just press the PS button for a few seconds.

BUILDING ANDROID ON THE ODROID-C1

A WALKTHROUGH FOR COMPILING KITKAT

by Nanik Tolaram

The ODROID-C1 has been out for quite some time and it is an interesting piece of hardware. In addition to the very affordable $35 price tag, Amlogic also provides the source code and binaries for Android KitKat. In this article I will walk through the process of building your own Android to be used in your C1 board.

Processor vendors such as Samsung and Amlogic provide their own method and style of build systems for running on their processor products. This can be a bit tiresome for developers as they will now need to maintain different scripts and Makefiles for different systems. There are also slight differences in terms of file layout compared to the U3, which I discussed in the April 2014 issue of ODROID magazine (`http://bit.ly/1vkwuYk`). In this article, we will look at the ODROID-C1 in more detail.

Prerequisites

The main prerequisites for building Android is to have a fast computer and sufficient RAM. It is recommended that you use a SSD hard drive to build Android because you will save a tremendous amount of time compared to using a normal IDE hard drive. Refer to the article published in April 2014 (`http://bit.ly/1vkwuYk`) for more information on the hardware requirements for building Android.

Toolchains

The toolchains are very different for the C1 when compared to the U3 or any other ODROID product. This is because it uses different compilers for compiling different parts of the Android system. Table 1 lists the different toolchains that will need to be downloaded.

Subsystem	Download URL	Description
u-boot	http://bit.ly/1KKQrvt	Bootloader for booting C1
kernel	http://bit.ly/1Uh1LEg	Linux kernel for Android

Table 1: Toolchains

Normally, for building an Android system you do not require different toolchains for building the different parts of the system, and internally the Android toolchain would be self sufficient. The C1 however requires separate toolchains for the kernel and for u-boot. Extract the toolchains and put it somewhere in your local drive. In my case I have it under

```
/media/SeagateInternal/Android/Hardkernel/Images-
Files/gcc-linaro-arm-linux-gnueabihf-4.9-2014.09_
linux/
```

and

```
/media/SeagateInternal/Android/Hardkernel/Images-
Files/gcc-linaro-arm-none-eabi-4.8-2014.04_linux/
```

Android Source

Like any other Android system, you will need to use the repo tool from Google, and you can follow the instructions in `http://bit.ly/1Syr1sf` to download the repo tool. After downloading the repo tool, you will need to use the following command (repo checkout code):

```
repo init -u https://github.com/hardkernel/android.
git -b s805_4.4.2_master
repo sync
```

The manifest file for the C1 will download the source for

3 different locations:

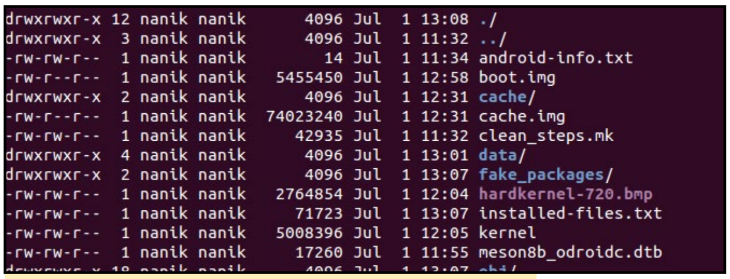
```
https://github.com/hardkernel
https://github.com/codewalkerster
https://android.googlesource.com
```

Building Android

Building an Android image for the C1 is straight forward, and you can use the buildC1.sh script listed in Table 2 below. Place the script in your Android source directory in order to make it easier to use. Make sure you change the PATH environment variable in the script to point to the correct local directory.

```
#!/bin/bash
export PATH=/media/SeagateInternal/Android/Hard-
kernel/ImagesFiles/gcc-linaro-arm-linux-gnueabi-
hf-4.9-2014.09_linux/bin:/media/SeagateInternal/
Android/Hardkernel/ImagesFiles/gcc-linaro-arm-none-
eabi-4.8-2014.04_linux/bin:$PATH
USE_CCACHE=1
CCACHE_DIR=/media/SeagateInternal/Android/ccache
source build/envsetup.sh
lunch odroidc-eng
make -j8  2>&1 | tee build.log
```

This build script will create a file named build.log, which is a log of the build messages that is often useful for troubleshooting purposes. After successful completion of the build process, you will see your out/target/product/odroidc directory as shown in Figure 1 below.



```
drwxrwxr-x 12 nanik nanik      4096 Jul  1 13:08 ./
drwxrwxr-x  3 nanik nanik      4096 Jul  1 11:32 ../
-rw-rw-r--  1 nanik nanik        14 Jul  1 11:34 android-info.txt
-rw-r--r--  1 nanik nanik   5455450 Jul  1 12:58 boot.img
drwxrwxr-x  2 nanik nanik      4096 Jul  1 12:31 cache/
-rw-r--r--  1 nanik nanik  74023240 Jul  1 12:31 cache.img
-rw-rw-r--  1 nanik nanik     42935 Jul  1 11:32 clean_steps.mk
drwxrwxr-x  4 nanik nanik      4096 Jul  1 13:01 data/
drwxrwxr-x  2 nanik nanik      4096 Jul  1 13:07 fake_packages/
-rw-rw-r--  1 nanik nanik   2764854 Jul  1 12:04 hardkernel-720.bmp
-rw-rw-r--  1 nanik nanik     71723 Jul  1 13:07 installed-files.txt
-rw-rw-r--  1 nanik nanik   5008396 Jul  1 12:05 kernel
-rw-rw-r--  1 nanik nanik     17260 Jul  1 11:55 meson8b_odroidc.dtb
```

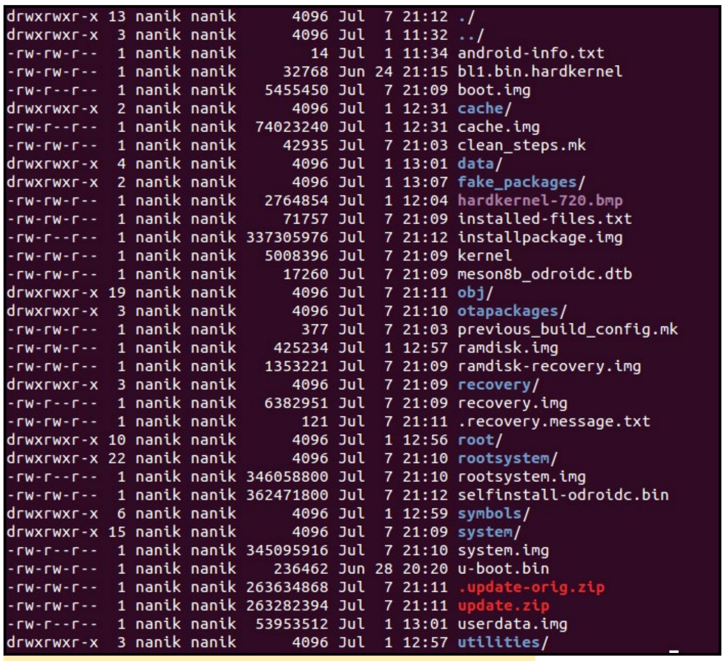
Figure 1 - contents of the build process

Packaging

The script in Table 3 will compile Android, but you still don't have the necessary files required for booting using the SD card. What we need to do now is to package all of the images in a format that can be copied over for booting purposes to the SD card. The step to do this is the same as Table 2, except that you will need to use a different target. For this step, you can use the buildC1Package.sh script shown in Table 3.

```
#!/bin/bash
export PATH=/media/SeagateInternal/Android/Hard-
kernel/ImagesFiles/gcc-linaro-arm-linux-gnueabi-
hf-4.9-2014.09_linux/bin:/media/SeagateInternal/
Android/Hardkernel/ImagesFiles/gcc-linaro-arm-none-
eabi-4.8-2014.04_linux/bin:$PATH
USE_CCACHE=1
CCACHE_DIR=/media/SeagateInternal/Android/ccache
source build/envsetup.sh
lunch odroidc-eng
make -j8  selfinstall  2>&1 | tee build.log
```

After successful completion of the packaging process, you will see your out/target/product/odroidc directory as shown in Figure 2.



```
drwxrwxr-x 13 nanik nanik      4096 Jul  7 21:12 ./
drwxrwxr-x  3 nanik nanik      4096 Jul  1 11:32 ../
-rw-rw-r--  1 nanik nanik        14 Jul  1 11:34 android-info.txt
-rw-rw-r--  1 nanik nanik     32768 Jun 24 21:15 bl1.bin.hardkernel
-rw-r--r--  1 nanik nanik   5455450 Jul  7 21:09 boot.img
drwxrwxr-x  2 nanik nanik      4096 Jul  1 12:31 cache/
-rw-r--r--  1 nanik nanik  74023240 Jul  1 12:31 cache.img
-rw-rw-r--  1 nanik nanik     42935 Jul  7 21:03 clean_steps.mk
drwxrwxr-x  4 nanik nanik      4096 Jul  1 13:01 data/
drwxrwxr-x  2 nanik nanik      4096 Jul  1 13:07 fake_packages/
-rw-rw-r--  1 nanik nanik   2764854 Jul  1 12:04 hardkernel-720.bmp
-rw-rw-r--  1 nanik nanik     71757 Jul  7 21:09 installed-files.txt
-rw-r--r--  1 nanik nanik 337305976 Jul  7 21:12 installpackage.img
-rw-rw-r--  1 nanik nanik   5008396 Jul  7 21:09 kernel
-rw-rw-r--  1 nanik nanik     17260 Jul  7 21:09 meson8b_odroidc.dtb
drwxrwxr-x 19 nanik nanik      4096 Jul  7 21:11 obj/
drwxrwxr-x  3 nanik nanik      4096 Jul  7 21:10 otapackages/
-rw-rw-r--  1 nanik nanik       377 Jul  7 21:03 previous_build_config.mk
-rw-rw-r--  1 nanik nanik    425234 Jul  1 12:57 ramdisk.img
-rw-rw-r--  1 nanik nanik   1353221 Jul  7 21:09 ramdisk-recovery.img
drwxrwxr-x  3 nanik nanik      4096 Jul  7 21:09 recovery/
-rw-r--r--  1 nanik nanik   6382951 Jul  7 21:09 recovery.img
-rw-rw-r--  1 nanik nanik       121 Jul  7 21:11 .recovery.message.txt
drwxrwxr-x 10 nanik nanik      4096 Jul  1 12:56 root/
drwxrwxr-x 22 nanik nanik      4096 Jul  7 21:10 rootsystem/
-rw-r--r--  1 nanik nanik 346058800 Jul  7 21:10 rootsystem.img
-rw-rw-r--  1 nanik nanik 362471800 Jul  7 21:12 selfinstall-odroidc.bin
drwxrwxr-x  6 nanik nanik      4096 Jul  1 12:59 symbols/
drwxrwxr-x 15 nanik nanik      4096 Jul  7 21:09 system/
-rw-r--r--  1 nanik nanik 345095916 Jul  7 21:10 system.img
-rw-rw-r--  1 nanik nanik    236462 Jun 28 20:20 u-boot.bin
-rw-rw-r--  1 nanik nanik 263634868 Jul  7 21:11 .update-orig.zip
-rw-rw-r--  1 nanik nanik 263282394 Jul  7 21:11 update.zip
-rw-r--r--  1 nanik nanik  53953512 Jul  1 13:01 userdata.img
drwxrwxr-x  3 nanik nanik      4096 Jul  1 12:57 utilities/
```

Figure 2- contents of the packaging process

After successfully running the buildC1Package.sh script, you can now flash the 'selfinstall-odroidc.bin' file to your SD card using the dd command.

Booting Up

The first time you boot your C1, it will go through a 2 step process:

Step 1 will boot the C1 into recovery mode because a new installation has been detected. All necessary partitions for Android will be created, and the correct image files will be installed to the new partitions.

Step 2 will reboot the C1 directly to Android.

Next Month

In next month's article, we will take a deeper look into the boot process of Android running on the ODROID-C1 as the boot up is slightly different than the U3. We will further explore u-boot in particular as this is the driving force behind the booting process.

LINUX GAMING

SUPER PUZZLE FIGHTER II TURBO

by Rob Roy

One of my favorite games is Super Turbo Puzzle Fighter II Turbo, which is a competitive Tetris-style game that is available for many different consoles. Using the popular GameStation Turbo image, available at http://bit.ly/1IALWzP, allows it to run on the ODROID platform with Xbox 360 controllers.

The game is best played with two players, where each player tries to keep their pit of gems from filling up, while simultaneously connecting blocks of gems of the same color in order to drop gems on their opponent. In the center of the screen are two anime-style fighters that throw punches, kicks and special moves that correspond to offensive and defensive Tetris actions. One player wins when the other player's gem chute is blocked from too many pieces.

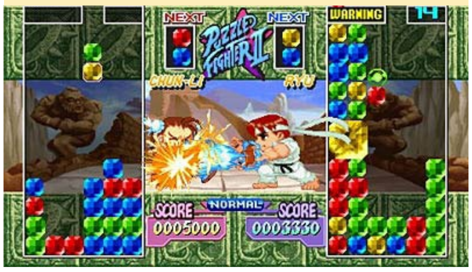
There are special spinning gems called "crash gems" that will make the connected patterns of same-colored gems disappear and pull a move on your opponent. Every 25th gem is a "diamond" that will make all gems on the entire board of the same color that it touches disappear, which results in an even bigger "gem drop".

I tested two different versions of Puzzle Fighter, the Game Boy Advance and the PlayStation Portable version, and while the GBA version has good gameplay, the PSP graphics are much better. To play STPFII, copy the PSP rom into the /home/odroid/ROMS/PSP folder of the GameStation Turbo image, then start up XBMC and use the ROM Collection Browser addon to add the ROM to your collection. Finally, make sure that both controllers are connected, start the game, and select the "Versus" option.

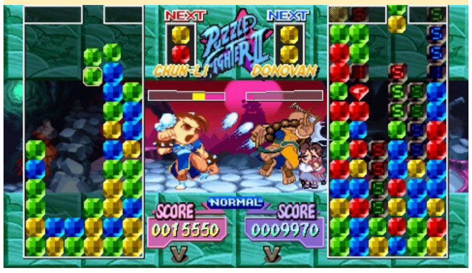
There are many characters to choose from, but the game was originally designed to give the advantage to Ken and Donovan, as they have the most difficult drop gem patterns, which means that your opponent will have a harder time recovering from a large "gem drop". It's best to try and create "super gems" made of many smaller gems, as these will yield the biggest moves when connected.

What makes this game different from Tetris is that having an empty "gem chute", which is usually desirable in regular Tetris, means that you don't have any offensive moves with which to confound your opponent. A good strategy is to keep a balanced board, because you have exactly one move to block your opponent's attacks by connecting your own gems. Any gems that you connect when you are being attacked are subtracted from their offensive move. Even though it may look like you are losing by having a nearly full gem pit, you are sometimes only one move away from pulling off a victory by chaining together several gem connections and dropping a giant pile of gems on your opponent! There are also several hidden characters to unlock, and we have had hours of fun playing Super Turbo Puzzle Fighter II in our living room. For more information about the game, please visit the STPFII Wikipedia page at http://bit.ly/1KM6o7D.

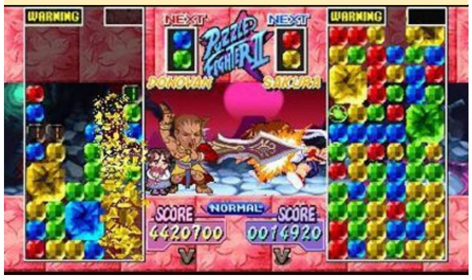
Player 2 pulls off an attack by cleverly connecting same-colored gems



Player 1 performs a counter-attack



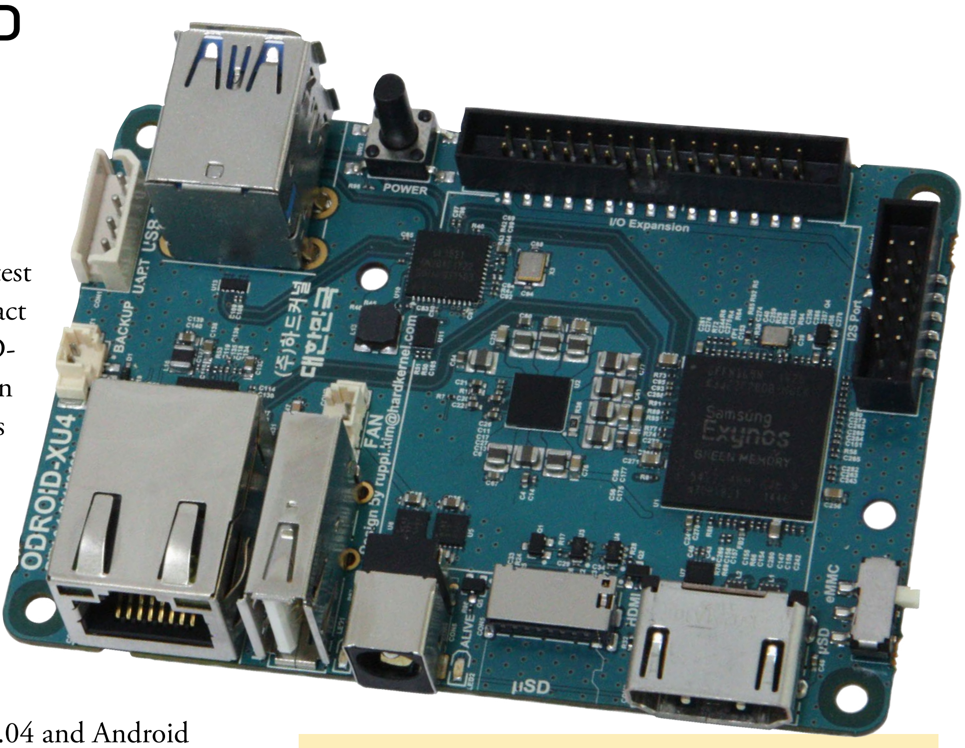
A devastating attack by Player 1, resulting in a large gem drop on Player 2



ODROID-XU4

A FRESH LOOK AT OUR NEWEST BOARD

by Justin Lee



A complete improvement that stays true to the Hardkernel philosophy, the XU4 packs an impressive punch on a really compact board

Hardkernel recently announced its latest ultra-fast octo-core 5422-based compact single board computer, the ODROID-XU4. Including several improvements upon the ODROID-XU3, the ODROID-XU4 is powered by ARM® big.LITTLE™ technology, the Heterogeneous Multi-Processing (HMP) solution. The ODROID-XU4 is a new generation of computing device with more powerful, more energy-efficient hardware and a smaller form factor. Offering open-source support, the board can run various flavors of Linux, including the latest Ubuntu 15.04 and Android 4.4 KitKat and 5.0 Lollipop.

By implementing eMMC 5.0 technology, USB 3.0 and Gigabit Ethernet interfaces, the ODROID-XU4 boasts amazing data transfer speeds, a feature that is increasingly required to support advanced processing power on ARM devices. This allows users to truly experience an upgrade in computing, especially with faster booting, web browsing, networking, and 3D games.

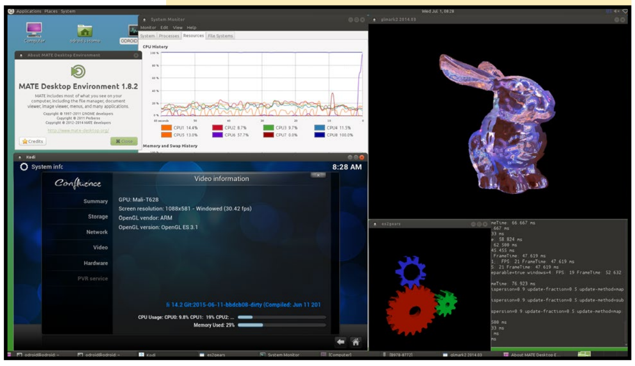
- **Samsung Exynos 5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core CPUs at 1.4GHz**
- **Mali-T628 MP6 (OpenGL ES 3.1/3.0/2.0/1.1 and OpenCL 1.1 Full profile)**
- **2Gbyte LPDDR3 RAM PoP stacked**
- **eMMC5.0 HS400 Flash Storage**
- **2 x USB 3.0 Host, 1 x USB 2.0 Host**
- **Gigabit Ethernet port**
- **HDMI 1.4a for display**
- **Size : 82 x 58 x 22 mm approx. (including cooling fan)**

OpenGL-ES 3.0

The ARM® Mali™-T628 MP6 GPU offers key API support for OpenGL ES 1.1, OpenGL ES 2.0 and OpenGL ES 3.0, OpenCL 1.1 Full Profile and Google RenderScript. Mali-T628 is the GPU of choice for use in the next generation of market-leading devices, optimized to bring breathtaking graphical displays to consumer applications such as 3D graphics, visual computing, augmented reality, procedural texture generation and voice recognition. You can download the full featured OpenGL ES and OpenCL SDK from ARM Mali Developer website for free!

The screenshot shows OpenGL-ES applications and the Kodi media player with Ubuntu 15.04 Mate desktop on the HMP enabled Kernel 3.10 LTS. The latest Kernel 4.2.0 RC1

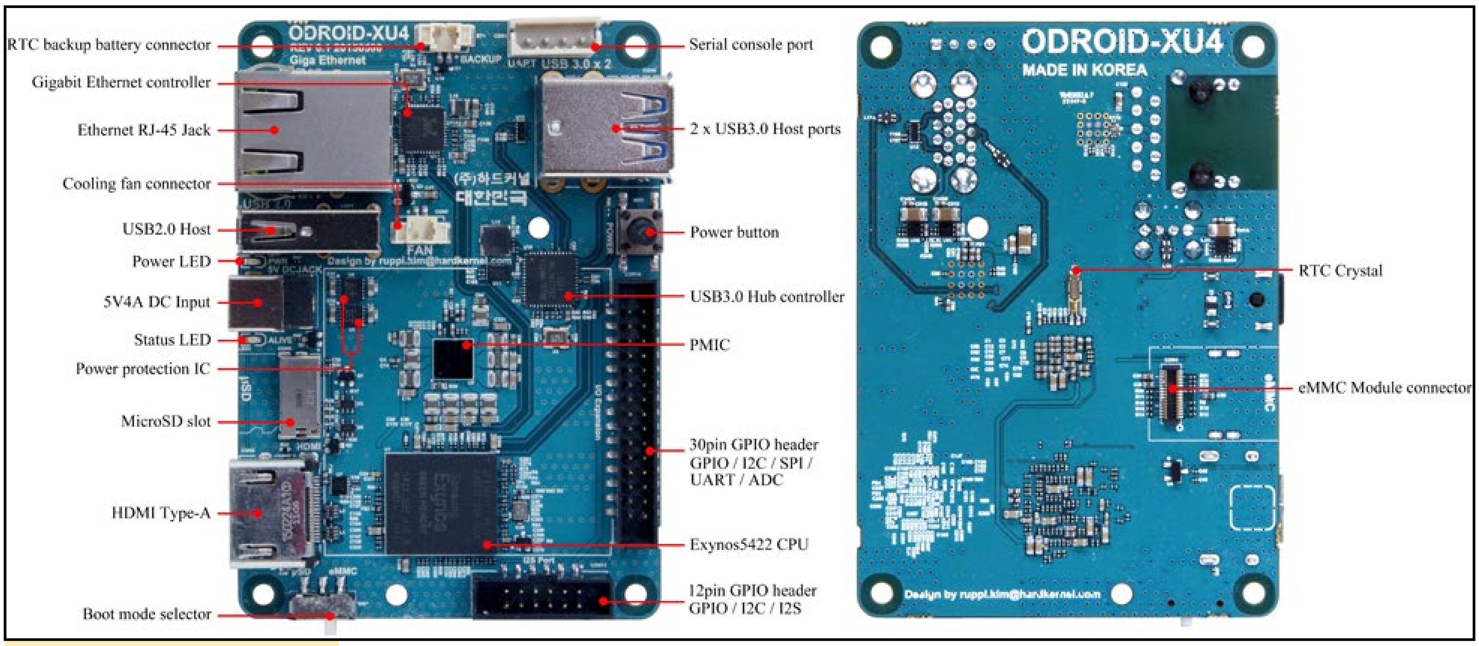
Ubuntu 15.04 Mate Desktop with Kernel 3.10



MATE includes most of what you see on your computer, including the file manager, document viewer, image viewer, menus, and many applications.
Copyright © 1997-2011 GNOME developers
Copyright © 2011 Perberos
Copyright © 2012-2014 MATE developers
http://www.mate-desktop.org/

CPU1 14.4% CPU2 8.7% CPU3 9.7% CPU4 11.5%
CPU5 13.0% CPU6 57.7% CPU7 0.0% CPU8 100.0%

8:28 AM
Video information
GPU: Mali-T628
Screen resolution: 1088x581 - Windowed (30.42 fps)
OpenGL vendor: ARM
OpenGL version: OpenGL ES 3.1
li 14.2 Git:2015-06-11-bbdcb08-dirty (Compiled: Jun 11 201
CPU Usage: CPU0: 9.8% CPU1: 19% CPU2: ...
Memory Used: 29%



Top and Bottom board details

also runs on the XU4. The source code is available from Hardkernel's GitHub at http://bit.ly/1N2WImp. This experimental Kernel 4.2 support the SMP 4 x A15 cores, USB 3.0, Gigabit Ethernet and some other basic features. However, HDMI, GPU, VPU(MFC), and HMP drivers are not yet available in Kernel 4.2, so it is useful only for headless applications.

Comparison

The ODROID-XU4 is fully software compatible with the XU3, but is more compact, more affordable, and more expandable. The ODROID-XU4 offers several improvements over the previous XU3 model:

Gigabit Ethernet
More Stable Dual USB 3.0 host ports
More compact PCB size
More IO ports (I2S/I2C/GPIO)

In order to lower the price and decrease the physical footprint of the board, several of the ODROID-XU3 features were removed:

No USB OTG
No DP
No Audio CODEC
No Power Monitoring Sensors

Links

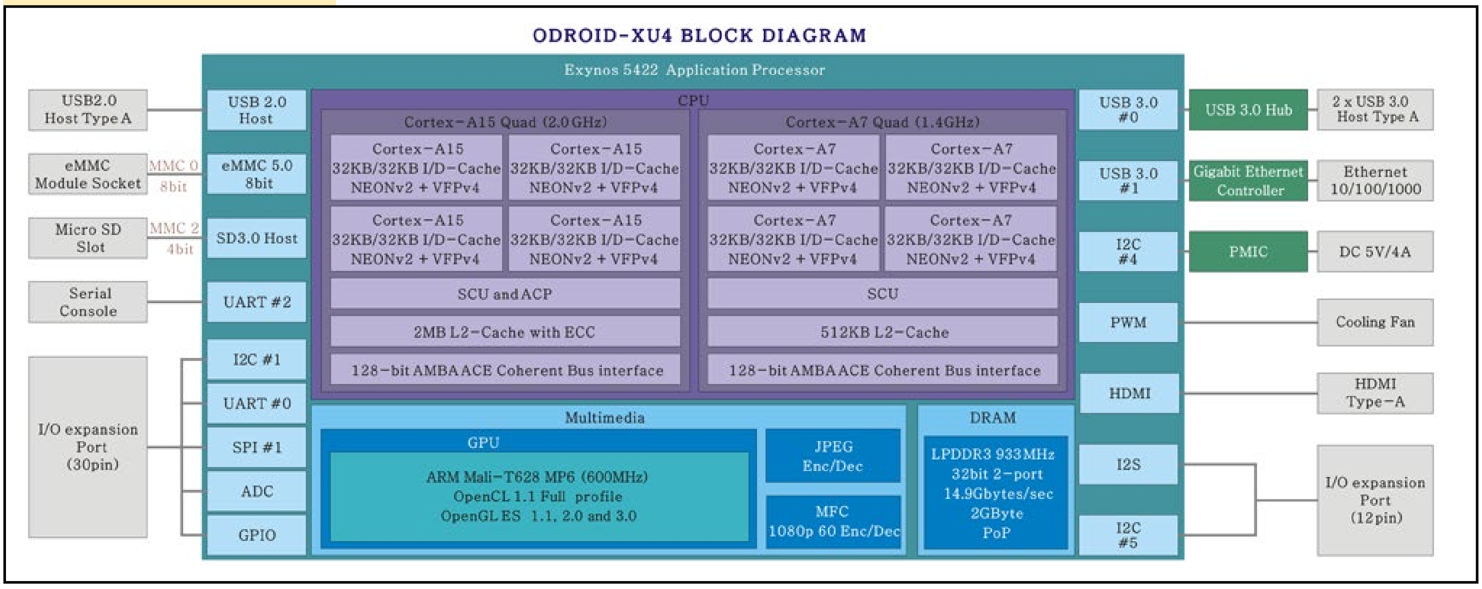
Wiki:
`http://bit.ly/1IF3Kyh`

Schematics:
`http://bit.ly/1VVNtL5`

PCB mechanical drawings (AutoCAD format):
`http://bit.ly/1OJqpK1`

Official Case Design File:
`http://bit.ly/1E3OJ3w`

The ODROID-XU4 is available for US$74 at `http://bit.ly/1fbE9ld`.



ODROID-XU4 Block Diagram

GRAILS
THE GROOVY VERSION OF RUBY ON RAILS

by Venkat Bommakanti

Have you ever wondered about the existence of a web application framework for the Java Virtual Machine (JVM) that enhances developer productivity? There is one: a 10-year veteran open-source framework called Grails! Being a mature framework, it's built using a powerful object-oriented dynamic language called Groovy for the Java platform, which is similar to Python and Ruby. Because the 20+ year old Java language was not originally developed for web-development, Groovy was invented to address the need for an efficient, non-cumbersome language to quickly develop web applications.

Due to their strengths, these technologies are lending themselves to extensive adoption in Internet Of Everything and Internet of Things solutions. This article is intended to help you get started with Groovy and Grails on the ODROID platform. An ODROID-U3 is recommended, because the 2GB memory requirement to build Grails is more than the 1GB RAM available on the ODROID-C1. It may be possible to create an installable package for a C1-class device, but the current software tools do not yet support it.

Requirements

1. An ODROID-U3 board, with an appropriate power adapter.
2. A Class 10 MicroSD or 8+ GB eMMC card with a micro SD card reader/writer, containing the latest U3-specific Lubuntu desktop image.
3. A network where the device has access to the Internet and the ODROID forums.

Preparation

Bring up the U3 with the latest Lubuntu desktop software. Expand the boot partition via the ODROID Utility. Reboot the device and update the system by selecting all the remaining relevant menus of the ODROID Utility, then reboot again.

GVM

The installation of Groovy is made easy through the Groovy enVironment Manager (gvm). It also helps with the management of parallel installed versions of multiple tools on many Linux systems. In addition, it makes it easy to switch between these versions.

Install gvm first using the command:

```
$ curl -s get.gvmtool.net | bash

Thanks for using

____/\\\\\\\\\\\\_/\\\________/\\\_/\\\\____________/\\\\_
 __/\\\//////////__\/\\\_______\/\\\_\/\\\\\\________/\\\\\\_
  _/\\\_____________\//\\\______/\\\__\/\\\//\\\____/\\\//\\\_
   _\/\\\____/\\\\\\\__\//\\\____/\\\___\/\\\\///\\\/\\\/_\/\\\_
    _\/\\\___\/////\\\___\//\\\__/\\\____\/\\\__\///\\\/___\/\\\_
     _\/\\\_______\/\\\____\//\\\/\\\_____\/\\\____\///_____\/\\\_
      _\/\\\_______\/\\\_____\//\\\\\______\/\\\_____________\/\\\_
       _\//\\\\\\\\\\\\/_______\//\\\_______\/\\\_____________\/\\\_
        __\////////////__________\///________\///______________\///__

Will now attempt installing...
Looking for a previous installation of GVM…
…
All done!
Please open a new terminal, or run the following in
the existing one:
  source "/home/odroid/.gvm/bin/gvm-init.sh"
Then issue the following command:
  gvm help
Enjoy!!!
```

Setup the gvm path using:

```
$ source "$HOME/.gvm/bin/gvm-init.sh"
```

Then, check the installation of gvm using the commands:

```
$ ls -ltr ~/.gvm
...
drwxrwxr-x 2 odroid odroid 4096 jul 11 13:46 ext
drwxrwxr-x 2 odroid odroid 4096 jul 11 13:46 crash
...

$ gvm help
Usage: gvm <command> <candidate> [version]
      gvm offline <enable|disable>

      commands:
              install   or i <candidate> [version]
              uninstall or rm   <candidate> <version>
              list    or ls   <candidate>
              use     or u    <candidate> [version]
              default   or d <candidate> [version]
              current   or c [candidate]
              outdated  or o [candidate]
              version   or v
              broadcast or b
              help    or h
              offline          <enable|disable>
              selfupdate       [force]
              flush            <candidates|broadcast|ar
chives|temp>

      candidate: asciidoctorj, crash, gaiden, glide,
gradle,
grails, griffon, groovy, groovyserv, jbake,
lazybones, springboot, vertx
      version: where optional, defaults to latest
stable if not
provided

eg: gvm install groovy
```

Groovy installation

Install latest Groovy Ver. 2.4.3, selecting 'Y' (Yes) to make this version the default:

```
$ gvm install groovy
==== BROADCAST ==================================================
   * 10/07/15: Springboot 1.3.0.M2 has been released on GVM.
#springboot
   * 09/07/15: Grails 3.0.3 has been released on GVM.  #grailsfw
   * 09/07/15: Vertx 3.0.0 has been released on GVM.  #vertx
=================================================================

Downloading: groovy 2.4.3

% Total   % Recvd % Xfrd  Average Speed   Time
Time  Time  Current
                        Dload  Upload  Total
Spent Left  Speed
  0    0      0       0       0       0       0       0
-:--:-- -:--:-- -:--:--      0
  0    0      0       0       0       0       0       0
-:--:-- 0:00:01 -:--:--      0
  0    0      0       0       0       0       0       0
-:--:-- 0:00:01 -:--:--      0
100 30.4M   100  30.4M       0       0  2136k       0
0:00:14 0:00:14 --:--:-- 2502k

Installing: groovy 2.4.3
Done installing!

Do you want groovy 2.4.3 to be set as default? (Y/n):
Y

Setting groovy 2.4.3 as default.
```

Check the installation using the commands:

```
$ which groovy
/home/odroid/.gvm/groovy/current/bin/groovy

$ ls -lsa /home/odroid/.gvm/groovy/current/
...
     4 drwxr-xr-x 8 odroid odroid  4096 mar 23 16:06
.
     4 drwxrwxr-x 3 odroid odroid  4096 jul 11 13:47
..
     4 -rw-r--r-- 1 odroid odroid  1167 mar 23 16:04
ANTLR-LICENSE.txt
     4 -rw-r--r-- 1 odroid odroid  1692 mar 23 16:04
ASM-LICENSE.txt
…

$ groovy -version
Groovy Version: 2.4.3 JVM: 1.8.0_33 Vendor: Oracle
Corporation OS: Linux
```

Try a sample "Hello World" app and time it:

```
$ time groovy -e 'println("Hello, World!")'
Hello, World!
```

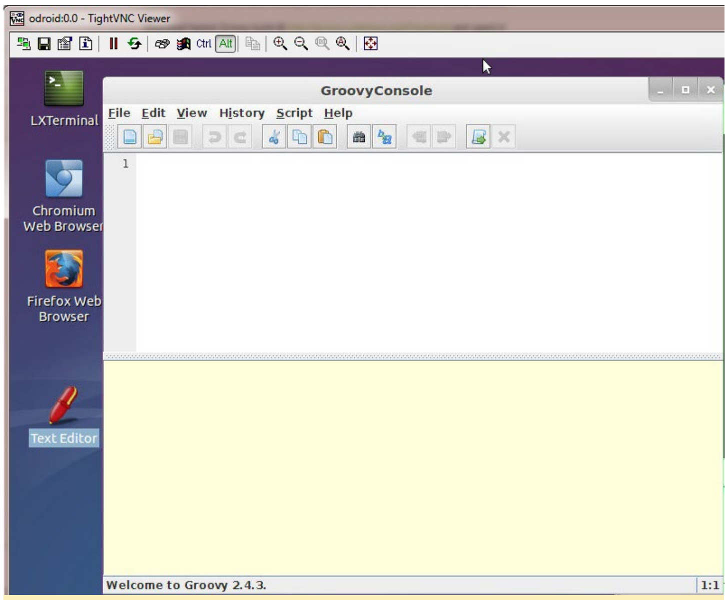
```
real    0m2.424s
user    0m2.335s
sys     0m0.500s
```

Although it works, it is rather slow! Next, check the installation of the Groovy Console:

```
$ which groovyConsole
/home/odroid/.gvm/groovy/current/bin/groovyConsole
```

Then, launch the Groovy console, which should display the screen as shown in Figure 1:

```
$ groovyConsole &
```



```
Welcome to Groovy 2.4.3.
```

Figure 1: Groovy console

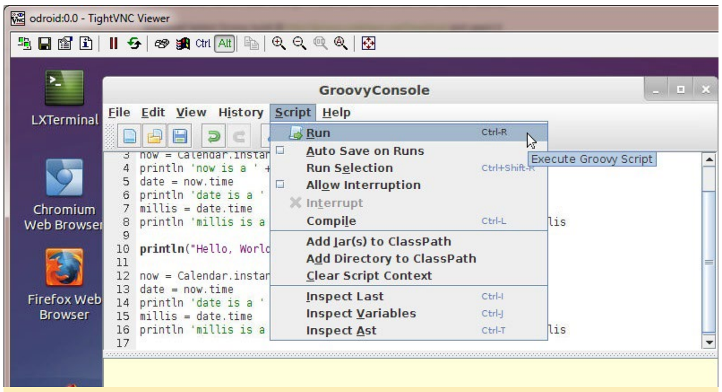
If there are black icons in console, or bold menu text, then change BPP mode in the boot.ini file of the boot media, to match the following, then save and reboot.

```
...
setenv m_bpp "24"
...
```

The use of the console as an Integrated Development Environment (IDE) can be tested by pasting the following code script snippet in the console code window:

```
println '........'

now = Calendar.instance
println 'now is a ' + now.class.name
date = now.time
println 'date is a ' + date.class.name + ' with value
' + date
millis = date.time
println 'millis is a ' + millis.class.name + ' with
value ' + millis

println("Hello, World!")

now = Calendar.instance
date = now.time
println 'date is a ' + date.class.name + ' with value
' + date
millis = date.time
println 'millis is a ' + millis.class.name + ' with
value ' + millis
```

Save the above code to a file by name: hello-world.groovy. From the console menu, click the Run option as shown in Figure 2.



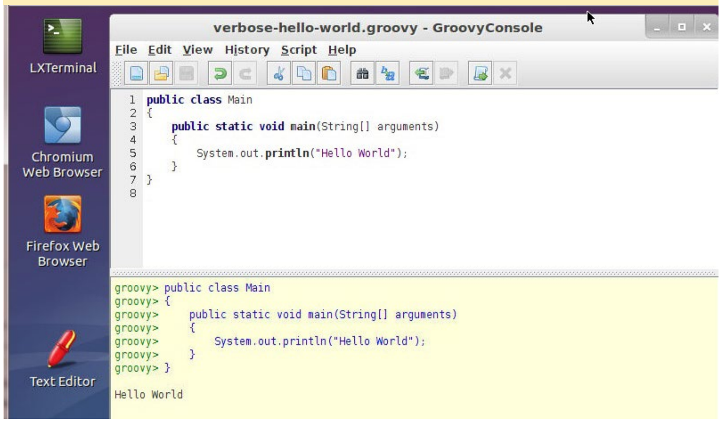
```
now = Calendar.instar
println 'now is a ' +
date = now.time
println 'date is a '
millis = date.time
println 'millis is a                lis

println("Hello, Worlc

now = Calendar.instar
date = now.time
println 'date is a '
millis = date.time
println 'millis is a                lis
```

Figure 2: Run Groovy script

Before we get ahead of ourselves, let us examine the power of Groovy. Consider the following simple object oriented program snippet, whose output is shown in Figure 3.

```
public class Main
{
    public static void main(String[] arguments)
    {
```

Figure 3: Run object oriented Groovy script


LXTerminal
Chromium Web Browser
Firefox Web Browser
Text Editor
verbose-hello-world.groovy - GroovyConsole
File Edit View History Script Help

```
public class Main
{
    public static void main(String[] arguments)
    {
        System.out.println("Hello World");
    }
}
```

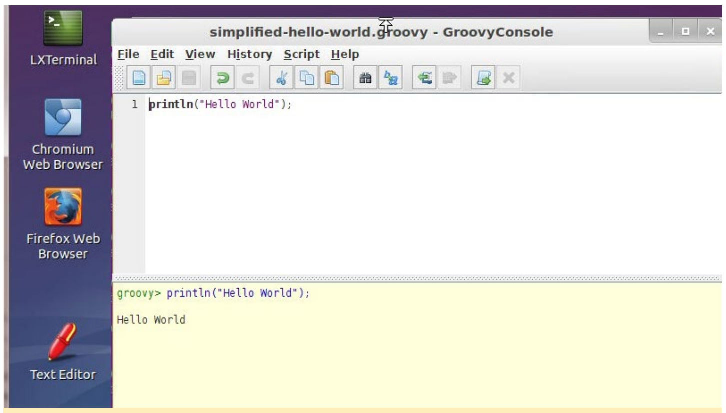
```
groovy> public class Main
groovy> {
groovy>     public static void main(String[] arguments)
groovy>     {
groovy>         System.out.println("Hello World");
groovy>     }
groovy> }

Hello World
```

```
        System.out.println("Hello World");
    }
}
```

Using the power of the language, the above snippet can be simplified to:

```
println "Hello World"
```

Note the similarity of this aspect of the Groovy language to a typical scripting language. Therein lies its power, as shown in Figure 4.


LXTerminal
Chromium Web Browser
Firefox Web Browser
Text Editor
simplified-hello-world.groovy - GroovyConsole
File Edit View History Script Help

```
println("Hello World");
```

```
groovy> println("Hello World");
Hello World
```

Figure 4: Run simplified Groovy script

Grails installation

Install the latest version of Grails (3.0.3) using the following command:

```
$ source "$HOME/.gvm/bin/gvm-init.sh"
$ gvm install grails
    ...
    Installing: grails 3.0.3
    Done installing!

    Do you want grails 3.0.3 to be set as default?
(Y/n): y

    Setting grails 3.0.3 as default.
```

Update path to Grails by editing your .bashrc file using this command:

```
$ medit ~/.bashrc
```

Then, add the following snippet:

```
# grails (current ver = 3.0.3) home
export GRAILS_HOME=$HOME/.gvm/grails/current/bin/
grails
export PATH=$PATH:$GRAILS_HOME/bin
```

Save the changes and update the environment variables:

```
$ source ~/.bashrc
```

Verify the Grails installation:

```
$ which grails
/home/odroid/.gvm/grails/current/bin/grails

$ grails --version
Java HotSpot(TM) Client VM warning: TieredCompilation
is disabled in this re| Grails Version: 3.0.3
    | Groovy Version: 2.4.3
    | JVM Version: 1.8.0_33
```

Create a sample Grails application using the command:

```
$ cd ~
$ grails create-app grails-sample
Java HotSpot(TM) Client VM warning: TieredCompilation
is disabled in this release
| Application created at /home/odroid/grails-sample
$ cd grails-sample/
$ ls -lsa
    ...
    4 drwxrwxr-x  5 odroid odroid 4096 jul 11 14:41 .
    4 drwxrwxr-x  4 odroid odroid 4096 jul 11 14:41
..
    4 -rw-rw-r--  1 odroid odroid 2012 jul 11 14:41
build.gradle
    4 drwxrwxr-x  3 odroid odroid 4096 jul 11 14:41
gradle
    4 -rw-rw-r--  1 odroid odroid   45 jul 11 14:41
gradle.properties
    8 -rwxrw-r--  1 odroid odroid 5080 jul 11 14:41
gradlew
    4 -rw-rw-r--  1 odroid odroid 2404 jul 11 14:41
gradlew.bat
    4 drwxrwxr-x 12 odroid odroid 4096 jul 11 14:41
grails-app
    4 drwxrwxr-x  5 odroid odroid 4096 jul 11 14:41
src
```

Note that this step fails if run on the ODROID-C1 due to the lack of free memory. The sample application can be started using the command:

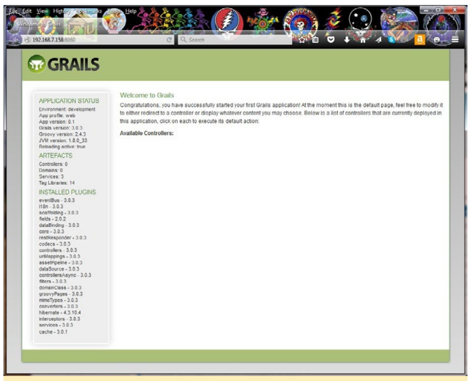
```
$ grails run-app
...
```

```
| Running application...
Java HotSpot(TM) Client VM warn-
ing: TieredCompilation is dis-
abled in this release.
Grails application running at
http://localhost:8080 in environ-
ment: development
```

Launch a web browser on another system on the network that can access the ODROID-U3, and point the browser to the address:

```
http://<u3-ip-address>:8080
```

You will see the sample Grails-rendered welcome homepage as shown in Figure 5.



APPLICATION STATUS
Environment: development
App profile: web
App version: 0.1
Grails version: 3.0.3
Groovy version: 2.4.3
JVM version: 1.8.0_33
Reloading active: true

ARTEFACTS
Controllers: 0
Domains: 0
Services: 3
Tag Libraries: 14

INSTALLED PLUGINS
eventBus - 3.0.3
i18n - 3.0.3
scaffolding - 3.0.3
fields - 2.0.2
dataBinding - 3.0.3
core - 3.0.3
restResponder - 3.0.3
codecs - 3.0.3
controllers - 3.0.3
urlMappings - 3.0.3
assetPipeline - 3.0.3
dataSource - 3.0.3
controllersAsync - 3.0.3
filters - 3.0.3
domainClass - 3.0.3
groovyPages - 3.0.3
mimeTypes - 3.0.3
converters - 3.0.3
hibernate - 4.3.10.4
interceptors - 3.0.3
services - 3.0.3
cache - 3.0.1

Welcome to Grails

Congratulations, you have successfully started your first Grails application! At the moment this is the default page, feel free to modify it to either redirect to a controller or display whatever content you may choose. Below is a list of controllers that are currently deployed in this application, click on each to execute its default action:

Available Controllers:

Figure 5: Sample Grails application's welcome homepage

The reader is encouraged to study these tools further in order to get proficient with the development of complicated web applications.

Additional resources

Visit the following links to learn more about Grails:

Grails
https://grails.org
Groovy
http://groovy-lang.org
GVM
http://gvmtool.net
Ubuntu Groovy
http://bit.ly/1OYrWwp
Sample code
http://www.grailsexample.net

USING BUILDROOT

CREATE A SIMPLE MEDIA PLAYER

by Garold Scantlen

I have a long employment history in R&D of microcontroller devices. In the mid 1980s, I developed a time-tagging device for audio recordings, which used a Motorola 68K-based 8-bit microcontroller. In the early 1990s, I developed a GPS real-time tracking, recording and mapping system using a 68332 u-ctlr together with a CF card, Tremble GPS, and Dataradio modem. Other projects of mine include robotics, automated control, data collection and processing devices utilizing various single board computers (SBCs). Some other SBCs that I've worked with are the AMD mini AT/XTs, PC/104 based, Altera FPGA boards, Arduino YUN, and I plan to use a Raspberry Pi in the future.

In this article, I present my latest project, which is a simple media player on the XU3 using BuildRoot. It all started when I bought a Samsung Galaxy Tab S to replace my failing laptop. I had never tried using an Android tablet as a replacement for an Ubuntu laptop, and I quickly realized the deficiencies and difficulties in customizing the Galaxy Tab S. While searching for solutions to the Galaxy, I discovered the Odroid-XU3. I thought that I could use this SBC to experiment with different customizations for my tablet, to include Ubuntu. I expect that someday I may get back to customizing my Galaxy, but for now I'm having too much fun with the Odroid-XU3.

I ordered the XU3 with a 32GB Ubuntu eMMC. After exploring its impressive capabilities while running Ubuntu, I purchased a 16G Sandisk Ultra micro SD. My first installation on the micro SD was the Android system. This went well, except for finding a reliable micro SD adapter for my desktop. I settled for a Targus micro SD USB dongle, then continued experimenting with Android Kitkat, Lollipop and Ubuntu, including rebuilds of Hardkernel's Linux kernel.

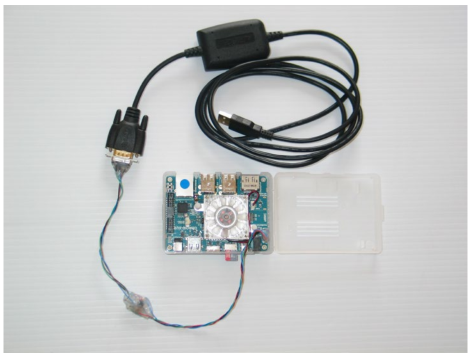
I then turned my focus to u-boot and re-discovered fastboot. Fastboot is not unique to Samsung/Android/smartphones. It is available in Hardkernel's u-boot package, and can be used with Ubuntu and other operating systems. U-boot provides useful features such as file and memory transfer, USB and ethernet interface, and even fdisk. A user interface to u-boot is available from the ODROID debug console using the USB-UART adapter whenever you interrupt the boot process. U-boot then provides a prompt from which fastboot can be executed. A client version of fastboot is then used on the workstation to copy partition blobs and filesystem images through the USB to the micro SD and eMMC.

You don't need a running or installed OS, just the running u-boot, a USB-

UART cable, and a micro USB cable. With my background in hardware, I built a debug console interface using 4 resistors and 2 transistors. I then experimented with ways of utilizing fastboot in order to remotely flash partitions. Fastboot has saved a lot of wear on my microSD, since I don't have to remove it every time I build a new root image.

I spent several days downloading and building the Android system from source. It was a long arduous process that failed several times, forcing me to start over each time. I did discover some methods to recover from failures, but it was a rough road to obtaining a working version of Kitkat. Fortunately, I was successful at building a functional Android 4.4.4 from Hardkernel's source repository.

I then searched, but failed to find, a repository for the Ubuntu source. Rebuilding Ubuntu was not a priority, since I was not interested in building another full OS. If I need to use Ubuntu, I can just install and update pre-built images from Odroid, whereas Android requires source rebuilds by downloading it from http://bit.ly/1hdvKPg (5422_4.4.4_master branch) in order to customize it to my needs.



Cable and download screen

Having used Linux From Scratch extensively back in the late 1990s, I was prepared to venture deeper into a do-it-yourself OS. I found 2 development systems worth mentioning: OpenEmbedded and BuildRoot. OpenEmbedded may be best for the top-end system development cases with a large product line to support. But my keep-it-simple philosophy led me away from its complex learning curve, and I decided to use BuildRoot. I did experiment some with OpenEmbedded, but was convinced that it was overly complicated for my needs. For a comparison between the two packages, a video presentation on the differences in OpenEmbedded and BuildRoot is available at http://youtu.be/3J-5SdDWbzM.

My first functional BuildRoot system used the Hardkernel's u-boot & kernel, and removed uInitrd. As for the BuildRoot system image, I made only a few obvious changes in "Target-options", as shown below. The micro SD still had the bootloaders, boot partition, boot.ini, kernel, dtb, and root partition from Ubuntu.

Luckily, the drivers required for bootup were built-in to that kernel, and I could eliminate initrd. I then reformatted the root partition, uncompressed the BuildRoot image onto it, installed the kernel modules, and modified boot.ini. The system that booted was limited to bare minimum functionality, and had only one keyboard input via the debug console. But I still got that tickled feeling you get when you throw it all together and it just works!

My BuildRoot configuration:

```
Target-options -> (ARM (little
endian))
                    Target Ar-
chitecture Variant (cortex-A7)
                    Target ABI
(EABIhf)
```

The XU3's boot.ini without uInitrd:

```
ODROIDXU-UBOOT-CONFIG
      setenv fdt_high
"0xffffffff"
      setenv sms-
macaddr "smsc95xx.
macaddr=00:1e:06:61:7a:39"
      setenv boot-
rootfs "console=tty1
console=ttySAC2,115200n8
root=UUID=e139ce78-9841-40fe-
8823-96a304a09859 rootwait ro"
      setenv bootcmd "fatload mmc
0:1 0x40008000 zImage; fatload
mmc 0:1 0x42000000 exynos5422-
odroidxu3.dtb; bootz 0x40008000 -
0x42000000"
      setenv mmcrootdev "root=/
dev/mmcblk0p2 rw"
      setenv mmcrootfstype
"rootfstype=ext4 rootwait"
      setenv videocon-
fig "drm_kms_helper.edid_
firmware=edid/1920x1080.bin"
      setenv bootargs "${boot-
rootfs} ${videoconfig} ${sms-
macaddr} ${mmcrootdev} ${mmcroot-
fstype}"
      boot
```

This list summarizes the next steps in building the XU3 media player.

- Add eudev, Buildroot's Dynamic /dev management

- Configure some tty consoles in /etc/inittab (tty1::respawn:/sbin/getty 38400 tty1)
- Configure the ethernet interface at /etc/network/interfaces
- Add ssh and rsync
- Add Xorg with vesa, keyboard, mouse, twm, and xterm
- Configure Xorg (/etc/X11/xorg.conf)
- Add ALSA audio and utils
- Add MESA 3D, openGL/ES/EGL, 3D demos, and other graphics libraries
- Add armsoc_drv (driver package

added to BuildRoot)

- Configure Xorg for armsoc (/etc/X11/xorg.conf)
- Add ffmpeg, mplayer, python interpreter with modules, and youtube-dl

Notes

Mplayer does not use 3D graphics, but it does use openGL/ES/EGL. Although I have successfully added the custom drive package, armsoc_drv, to support these graphics, full hardware acceleration for 3D graphics may require other Xorg drivers/libs to be built inside BuildRoot. However, there are licensing issues with the Mali GPU DDK, which is required to build a complete driver stack. The source code for the Mali GPU X11 display drivers are available at http://bit.ly/1MEj0gj.

Conclusion

More detailed instructions, along with configuration files for the XU3 MPlayer software, are available on my GitHub page at http://bit.ly/1Ii7Ecx. These instructions include procedures for adding the Armsoc driver and the debug console cable. Contributions to this project are greatly appreciated. Please email me from Github or post a new thread on the ODROID forums at http://forum.odroid.com. The BuildRoot use manual is available at http://bit.ly/1VVhgDE.

I plan to upload other instructional notes (.ins files) for various installs and builds to my GitHub repository. Some of them include builds of HPC clusters on Infiniband with GPU computes.

XU3 running GLXGears and YouTube Video

ODROID-C1 USER MANUAL

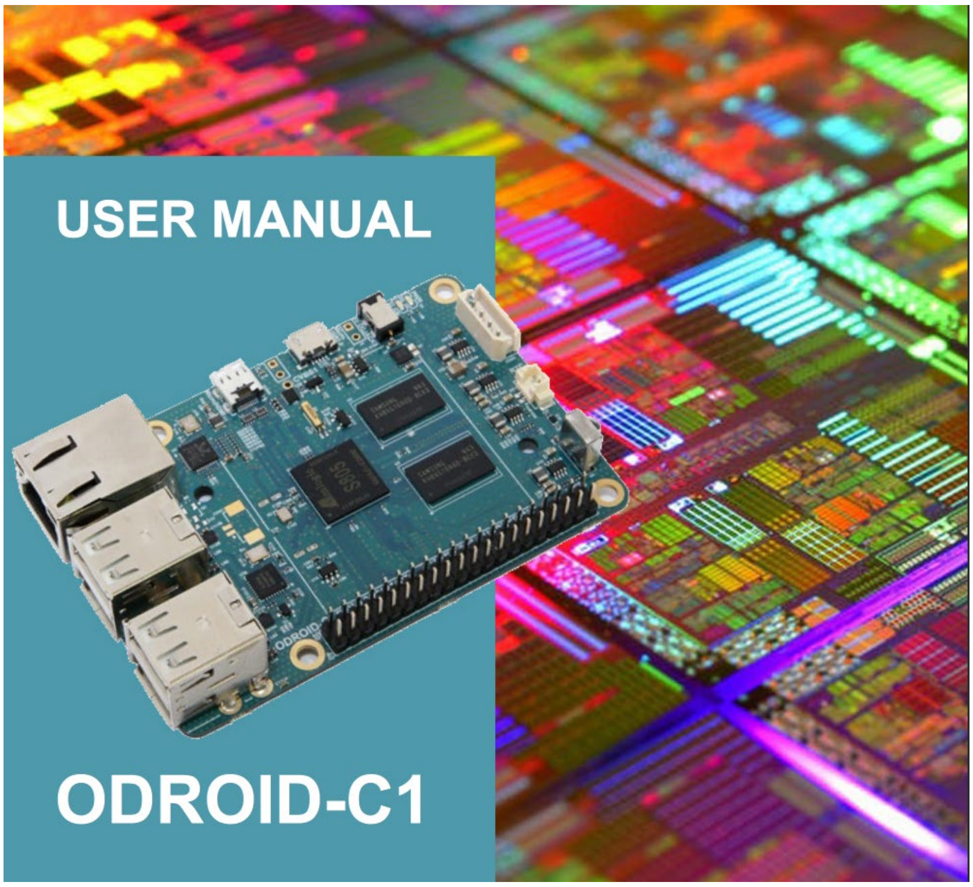
A GUIDE FOR ALL EXPERTISE LEVELS

by Rob Roy

The official user manual for the ODROID-C1 was recently released on the Hardkernel website, and is available for direct download at http://bit.ly/1K2NvMm, via the forums at http://bit.ly/1TQEgoK, and on the Google Play Store at http://bit.ly/1WA6vGV.

The ODROID-C1 is one of the most powerful low-cost Single Board computers available, as well as being an extremely versatile device. Featuring a quad-core AmLogic processor, advanced Mali GPU, and Gigabit ethernet, it can function as a home theater set-top box, a general purpose computer for web browsing, gaming and socializing, a compact tool for college or office work, a prototyping device for hardware tinkering, a controller for home automation, a workstation for software development, and much more.

Some of the modern operating systems that run on the ODROID-C1 are Ubuntu, Android, Fedora, ARCHLinux, Debian, and OpenELEC, with thousands of free open-source software packages available. The ODROID-C1 is an ARM device, which is the most widely used architecture for mobile devices and embedded 32-bit computing. The ARM processor's small size, reduced complexity and low power consumption makes it very suitable for miniaturized devices such as wearables and embedded controllers.



STEPMANIA
DANCING ENTERTAINMENT

by Oliver Schmitt

Karaoke, using the ultrastardx package, delivers a lot of musical fun with the ODROID platform. However, there are more music software packages available to make it a true musical entertainment machine. In this article, you will learn about the rhythm software called Stepmania, along with the necessary steps to get it running on the ODROID. I tested it using Debian Wheezy / Jessie and Ubuntu on both an ODROID U2 and a U3.

Figure 1 - Stepmania gameplay

Overview

Stepmania is a rhythm video game similar to Konami's arcade classic Dance Dance Revolution. While music is playing, arrows scroll upwards on the screen. The target of the game is to push buttons in time so that it fits the rhythm of the music perfectly. The game has single and multiplayer mode, so you can play to achieve a new topscore and/or versus one of your friends. I have confirmed that both modes work well. The game can be played either using standard inputs like a keyboard and joypad, or with special inputs like dance pads, so that it is not only a fun game but also a kind of body workout while listening to music - what a nice combination!

Stepmania uses OpenGL, so it will not work at a decent speed on ODROIDs without the use of @lunixbochs' well-known OpenGL wrapper. While the latest version of Stepmania (5.x) implements some OpenGL calls which are not compatible with it, version 3.9, which is used in this article, works very well.

Figures 2 and 3 - Single and Versus gameplay

Select Style
Don't hold back!
26
SINGLE
VERSUS
COUPLE
DOUBLE
SOLO
Select a play style.
VERSUS
2 players.
Each player plays on 4 panels.
Available options depend on how many players are joined.
16
SINGLE
1 player.
Each player plays on 4 panels.

Preparation

At a minimum, you will need to install the following development packages:

```
$ sudo apt-get install git \
  automake liblua5.1-0-dev \
  libmad0-dev libgtk2.0-dev \
  libsdl1.2-dev alsa-oss
```

As mentioned previous, you will also need OpenGL functionality. If you have added @meveric's repository to your system, simply install the following packages.

```
$ sudo apt-get install \
  libgl-odroid libglues-odroid
```

If you did not include his repository, you can download the individual deb files instead:

```
$ wget -O libgl-odroid.deb \
  http://bit.ly/1KEoF6O
$ wget -O libglues-odroid.deb \
  http://bit.ly/1VRydyN
$ sudo dpkg -i \
  libgl-odroid.deb \
  libglues-odroid.deb
```

Otherwise, you can compile and install lunixbochs glshim and glues manually using the code on GitHub at http://bit.ly/1DeHTNW.

If you're using Ubuntu, you will need some symlinks to get it working properly:

```
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/mali-egl/libmali.so \
  /usr/lib/arm-linux-gnueabihf/
libEGL.so.1.4
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/libEGL.so.1.4 \
  /usr/lib/arm-linux-gnueabihf/
libEGL.so.1.0
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/libEGL.so.1.0 \
  /usr/lib/arm-linux-gnueabihf/
libEGL.so.1
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/libEGL.so.1 \
  /usr/lib/arm-linux-gnueabihf/
libEGL.so
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/mali-egl/libmali.so \
  /usr/lib/arm-linux-gnueabihf/
libGLESv1_CM.so.1.1
$ ln -sf /usr/lib/arm-linux-
gnueabihf/libGLESv1_CM.so.1.1 /
usr/lib/arm-linux-gnueabihf/lib-
GLESv1_CM.so.1
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/libGLESv1_CM.so.1 \
  /usr/lib/arm-linux-gnueabihf/
libGLESv1_CM.so
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/mali-egl/libmali.so \
  /usr/lib/arm-linux-gnueabihf/
libGLESv2.so.2.0
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/libGLESv2.so.2.0 \
  /usr/lib/arm-linux-gnueabihf/
libGLESv2.so.2
$ ln -sf /usr/lib/arm-linux-gnue-
abihf/libGLESv2.so.2 \
  /usr/lib/arm-linux-gnueabihf/
libGLESv2.so
```

Compiling

Clone the repository and navigate to its directory:

```
$ git clone git://github.com/\
lunixbochs/stepmania-3.9
$ cd stepmania-3.9
```

Generate the configuration script:

```
$ ./autogen.sh
```

On Ubuntu and Debian Jessie, I got some error messages, just do as they suggest and enter the following command:

```
$ automake --add-missing
```

Some small changes in the configuration file need to be done, because it might not recognize liblua correctly otherwise. It will also try to use an ancient ffmpeg version, so we deactivate that, which will prevent optional videos from working inside the game.

In the created configure script comment out the exit command after the message you see below, which should be around line 6300 (depending on your system) and add the following two lines

```
echo
echo "*** liblua is required to
build StepMania; please make sure
that"
echo "*** it is installed to con-
tinue the installation process."
#     exit 1;
LUA_CFLAGS="-I/usr/include/
lua5.1"
LUA_LIBS=-llua5.1
```

Deactivate ffmpeg support around line 6900 (again, the line number depends on your system) by changing variable have_ffmpeg to "no":

```
FFMPEG_CFLAGS=$pkg_cv_FFMPEG_
CFLAGS
FFMPEG_LIBS=$pkg_cv_FFMPEG_LIBS
{ $as_echo "$as_me:${as_lineno-
$LINENO}: result: yes" >&5 $as_
echo "yes" >&6; }
have_ffmpeg=no
```

After these changes have been made, configuration should work fine. Ignore the error message about liblua, which is expected but not critical anymore after those changes above.

```
$ ./configure
```

Now check that -lX11 is added to LIBS in the file src/Makefile. The line should look like this:

```
LIBS = -lX11 -ldl  -lpng -lz -lm
-ljpeg -lz  -lpthread
```

Some modifications to the source need to be done:

```
$ wget -O patch.txt http://paste-
bin.com/raw.php?i=dXRCZn4r
$ patch -p0 < patch.txt
```

After these changes, you may compile and install Stepmania. On Debian, you might want to use checkinstall -D instead:

```
$ make -j5
$ sudo make install
```

Additional steps

Before you launch the game, you will also need graphics and songs. Download the x86 version of StepMania and extract that data from it.

```
$ mkdir ~/.stepmania-3.9
$ cd ~/.stepmania-3.9
$ wget -O StepMania-3.9a.tar.gz \
  http://bit.ly/1SQUmc7
$ tar xf StepMania-3.9a.tar.gz
$ mv StepMania-3.9/* .
$ rm StepMania-3.9a-linux.tar.gz
GtkModule.so stepmania
$ wget -O \
  Data/StepMania.ini \
  http://pastebin.com/\
  raw.php?i=132SbpLB
```

Afterwards, you still need some songs, which may be downloaded from any number of websites from the huge Stepmania community. Download your songs to the folder ~/.stepmania-3.9/Songs. Then, launch the game with the following script to make use of the OpenGL-wrapper:

```
#!/bin/bash
export LD_LIBRARY_PATH=/usr/lo-
cal/lib
aoss /usr/local/bin/stepmania
```

On all of the systems that I tested, there was a problem quitting the game. Although the window closes fine, the process remains and uses most of the CPU. The following script will launch Stepmania and automatically kill the process when you quit the game.

```
#!/bin/bash
export LD_LIBRARY_PATH=/usr/lo-
cal/lib
aoss /usr/local/bin/stepmania &
processid=$!
while true
  do
  sleep 1
  stepmaniawindow=`xwininfo -tree
-root | grep stepmania`
  if [ "$stepmaniawindow" == "" ]
    then
      break
    fi
  done
kill -9 $processid
```

Gameplay

Stepmania is highly customizable. You may, for example, change the themes, visualizations and noteskins to fit your personal taste. The most important part, of course, is your collection of songs, and there are lots of public domain songs online. For a beginner, I would recommend starting with songs that support multiple difficulties because you really need some training to get started. Another aspect you should pay attention to is whether the songs were made for your specific version of

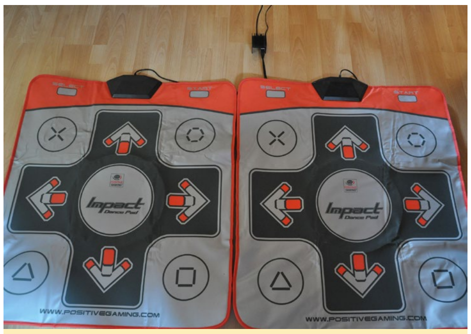


Figure 4 - Stepmania dance pads

Stepmania (3.9), and whether they are designed for keyboard or for dance pads.

There are a lot of different dance pads available for use with Stepmania, from hard metal to cheaper soft plastic pads. They are usually connected by USB and are recognized as joypads, so they should work out of the box. I use the two soft pads that can be seen in the photo, and am very satisfied with them. They are definitely a good start and already give me a true arcade feeling. Have fun dancing to the music on your ODROID.

For comments, suggestions, or questions about Stepmania, please visit the forum thread at http://bit.ly/1UcFpUk.

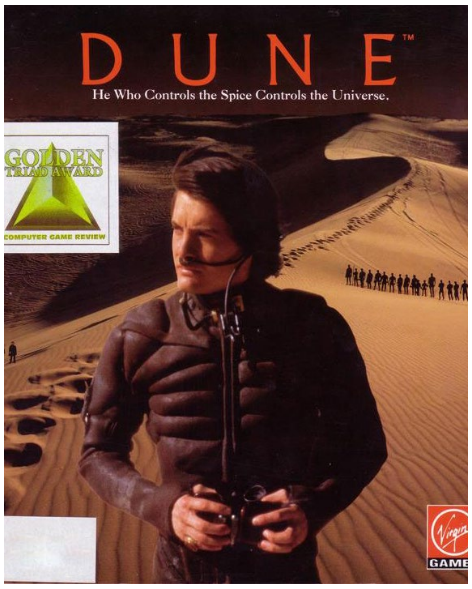
Always dance like no one is watching!

ODROID Magazine is now on Reddit!
ODROID Talk Subreddit
http://www.reddit.com/r/odroid

LINUX GAMING

RARE GAMING GEMS - PART 2

by Tobias Schaaf



Last month, I introduced the game Millennia – Altered Destinies, which is a very nice DOS game and a rare gaming gem. This time I want to talk about one of my favorite games called Dune. Similar to Millennia – Altered Destinies, it is a DOS game, but was also released for many other platforms, such as the Amiga and Sega CD. Many people know about Dune 2, the grandfather of all real time strategy games, but not so many know about the first Dune game, which was more like an adventure game, although it had a lot of strategy elements as well.

Figure 1 - Dune 2 was the predecessor of the famous Command and Conquer series from Westwood Studios

Dune 2 was a mixture of visual novel, adventure game, strategy game and some economics. I first played the game on the Amiga, but recommend the DOS CD version, since it had voice acting of all characters, as well as beautifully pre-rendered animations that appear when flying an Ornithopter or riding a Sand-worm. Besides that, the game offers a awesome soundtrack which was later released as a standalone CD version called "Dune: Spice Opera".

Installing

Since this is a DOS game and not specifically made for Linux, you can't just download a package and start. I use DOSBox to start the game and adjusted some options to make it work nicely. First of all, you need the CD or floppy version of Dune, although the CD version offers much better gameplay than the floppy version. Here is a step-by-step guide on how to install and start the game based on my ODROID GameStation Turbo image (although it should work the same with the Ubuntu image from HardKernel if you have my all/main and all/testing package lists activated as well):

```
$ sudo apt-get install dosbox-odroid libgl-odroid
```

Start DOSBox once to create the default config file but exit it right away. Open /home/odroid/.dosbox/dosbox-SVN.conf in a text-editor and change the following lines, after which you can start DOSBox from your menu.

```
[sdl]
fullscreen=true
fullresolution=1920x1080 (or any other resolution you use)
output=overlay
[render]
 frameskip=3
scaler=none
[cpu]
core=dynamic
cputype=auto
cycles=auto
cycleup=200
cycledown=200
```

If you prefer to use scalers to enhance the picture quality, you won't have the game fully scaled to 1080p, since the highest scaler is 3x which means 3 times the original game size. Because this game was originally in 320x240, that means 960x720 is the resulting picture, which can be somewhat small if you have 1080p. I prefer to use overlay with no scaler which enlarges the picture to the full desktop resolution. If you want to use scaler it might also be good to play it in Window mode. You can even choose OpenGL as an output driver since glshim is build into DOSBox. But before I did that, I created a folder where to place my games later:

```
$ mkdir DOS
```

I copied over the ISO from Dune and placed it into a folder called CDs on my ODROID. To make things easier, I added the following lines to the end of the DOSBox configuration file, so I don't need to type them every time I want to play the game:

```
[autoexec]
mount c: /home/odroid/DOS
c:
imgmount d: /home/odroid/CDs/
Dune.iso -t iso
```

Now the game is completely prepared and the emulator can be launched. The folder DOS will automatically mounted as my drive C:, and the CD will mounted as D: as a CD-ROM drive. Install the game as usual under DOS and start it.

Getting started

The game starts with a very nice introduction that includes the starting scene of the 1984 movie Dune from David Lynch, and later shows different scenes of the game itself, using 3D renderers as well as presenting some short conversations with different characters within the game. The entire introduction with all logos and scenes including the briefing before the game starts takes about 10 minutes and can be seen on Youtube at https://youtu.be/ATpH0aVH7lA

Figure 2 - An in-game video cut from the 1984 Dune movie

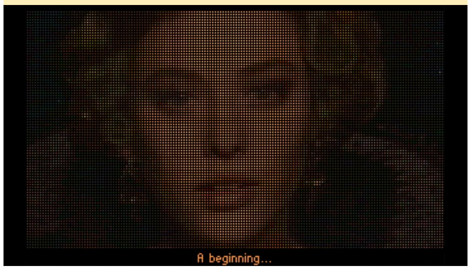


Figure 3 - A rendered flight scene from Dune

The CD version was a very good improvement of the original game. While the game graphics mostly stayed the same, some added features really improved the game, such as the very good voice acting of all characters and the new travel mode when you are in an Ornithopter or on a Sandworm. You play the game as Paul (Muad'Dib) Atreides and follow his story, travels and decisions. It's not that close to the original story of the book or movie version, but close enough to remember the characters if you have ever watched the movie or read the book.

You are send on the planet Dune to harvest the famous "spice" for the Emperor Shaddam IV. He will ask you for a shipment of spice every couple days which you have to send him, or you'll see a nice game-over scene where the emperor's troops arrive and kill you for disobeying his orders.

For this you need to make friends with the Fremens who will work for you, and in time, can take one of three occupations: harvesting spice, train for army, or ecologists. While at first you only can get Fremens to work for you as spice collectors, later in the game you can train them as your Army and eventually attack the Harkonnens, who are the enemies of your house and also stationed on Dune.

Gameplay

You play the game as Paul Atreides and travel from place to place speak with people. You can give them orders and follow the story as it progresses.

What sounds relatively simple soon becomes more and more complex. At first you only find a few sietches and have only a few Fremen you can interact with. This makes it relatively easy. You just tell them to harvest spice, and if you're lucky, you can send them out to find equipment to be more productive.

Figure 4 - Paul Atreides, main character and protagonist of this game

While Fremen can easily harvest spice by themselves, it's slow and takes rather long. So giving them a spice Harvester (which can be found in different sietches) will highly increase their productivity, but it also attracts Sandworms which eventually will try to attack the harvester. So you also need an Ornithopter to guard your spice Harvester. While this is easy with 5 to 10 troops of Fremen, it will get a lot harder when you have 20 or more troops. Later in the game, you can have military troops and ecologists as well. You need to train them and give them different commands on what tasks they should do.

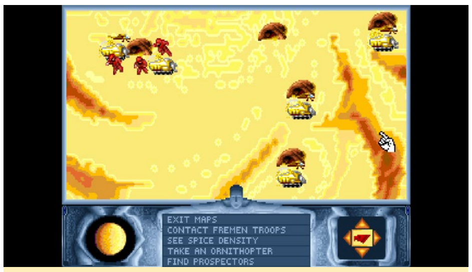


Figure 5 - Organizing troops, harvesting spice and military troops at training early in the game

Harvesting

As I said earlier, one of your main tasks in the game is to harvest spice. The Emperor will come to you every couple of days and ask for a new shipment of spice. This amount will increase each time he asks for a new shipment, so having a good working spice production is vital. Eventually, you will be able to buy equipment and weapons at smugglers' villages, who will also ask for spice as a compensation.

Early in the game, you meet a troop of spice prospectors who can prospect gray areas on the map to determine how much spice is in the area. You get a map from them which will show you the spice density of an area. The spice density map

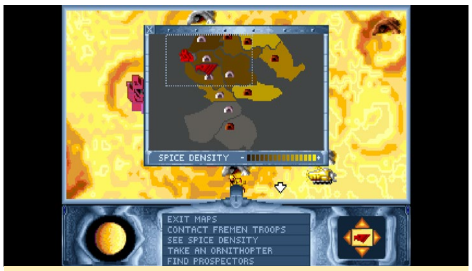


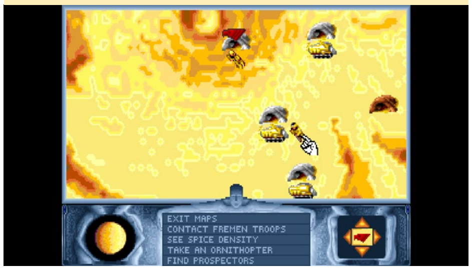
Figure 6 - An early spice density map. Yellow fields are rich on spice, on brown there is rarely spice to find. Gray areas need to be prospected first

is very helpful in your quest. Not only does it show you where you have to send your harvesters, if you look closely, you will find hidden sietches between the spots where you haven't discovered anything yet. It also shows you the range at which you can communicate with your troops. It's always good to harvest on a field until it turns brown and then send your troops to the next field. It is also beneficial to let multiple troops harvest on one field and therefore increasing efficiency. You can equip troops with a spice harvester to increase their efficiency on harvesting spice, but the harvester will attract sand worms that attack your harvesters and troops. Therefore, you should also add an Ornithopter as well, so they can spot sand worms and protect them from being attacked.

Telepathy

At first, you have to travel to each Fremen troop to give orders. This gets more difficult the more people you have, so while progressing in the story, you get the ability to get in contact with troops in the nearby areas depending on your telepathy ability.

Figure 7 - Sitting in the upper right corner, I can still contact all troops that are within the range of my telepathy shown by the gray sietches

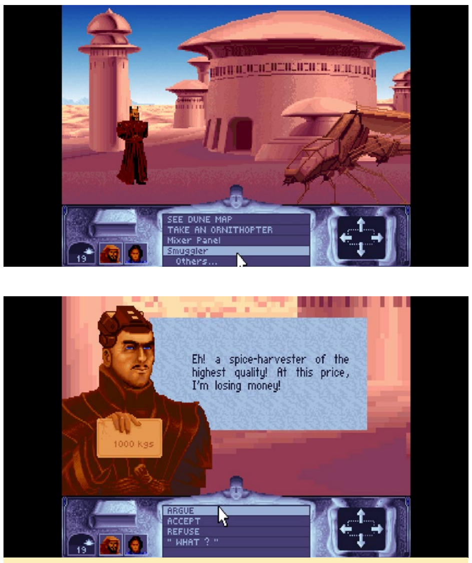


Telepathy makes it much easier to organize your troops. You can also see the range of your telepathy on the spice density map. As the game progresses, you get better, and can contact troops within a wider range, and doing the right thing at the right time even allows you to gain telepathy strong enough to communicate over the entire planet.

Villages

In time, you will get to a point where you can't find new equipment for all your troops anymore, and you have to find an alternative way to get it. You hear rumors of villages with where smugglers are supposed to live, and after some time you will find your first village and also some smugglers.

There are different smugglers on the planet, and each have different advantages and offer different prices and goods. Later on, you can also buy weapons from them. After you have purchased the items you want, you have to pay for them in your palace by speaking to Duncan Idaho, who in this game is responsible for watching over the spice production and paying bills, either to the smugglers or to the Emperor to satisfy his demands.



Eh! a spice-harvester of the highest quality! At this price, I'm losing money!

Figures 8 and 9 - A smuggler in a village haggling about the price of an spice harvester

After you have bought items from the smugglers ,you have to send your Fremen troops to a nearby sietch and tell them to search for equipment. They will then go to the village and grab what you bought. There is no way to transport items from one place to another, except for maybe an Ornithopter later in the game.

Villages are shown differently on a map. They do not take as much space on the map as a sietch does, and are normally hard to find. They are also shown differently on the map, appearing as little red spots rather than brown sietches or blue Harkonnen fortresses.

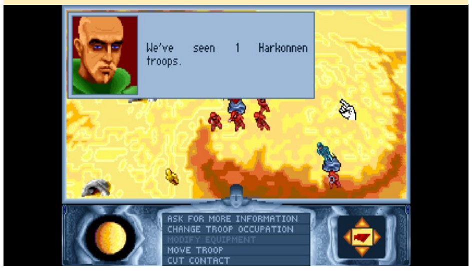
Exploring

Some of the sietches that you need to visit are not yet on your map, and you only receive general directions as to where to find these hidden sietches, like north-east, westwards, or south-west. Sometimes this is necessary in order to progress in the story, and sometimes it's just a way to find new sietches and therefore new troops and new spots where to harvest. Although it's rarely critical to the game, it's a major part of the fun. Finding new sietches can actually become a very nice pastime, where you try to find all of the hidden sietches and villages. New sietches mean more spice, more troops and often more free equipment. Therefore you should get used to reading the spice density map to see where more sietches could be.

Fighting

Eventually, you will get attacked by the Harkonnen and will also reach a point where you can't expand without pushing back the Harkonnen. There-

Figure 10 - Troops report from their espionage mission


We've seen 1 Harkonnen troops.

fore, you have to send your Troops on espionage missions, trying to find new Harkonnen fortresses and discovering how many troops they have in a fortress and what skills and weapons they have.

You normally should start far away from the Harkonnen palace and attack easier fortresses. After you have won a battle, your troops will transform the Harkonnen fortress into a sietch, which means that you can send harvesters there to get more spice. When your troops attack a fortress, you can go there to support them and give them a morale boost. You can not fly there, but have to use a sand worm.

After you have won a battle, you can

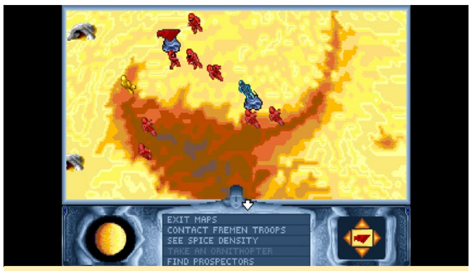


Figure 11 - Troops fighting against Harkonnen fortress

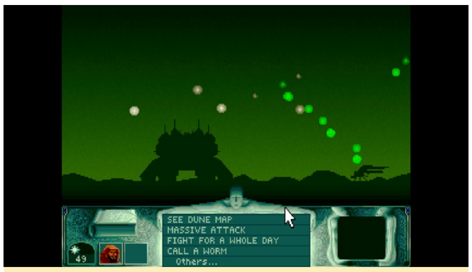


Figure 12 - A battle scene with nice effects of explosions in the sky

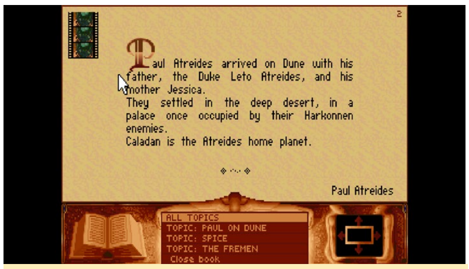
enter the Harkonnen fortress, where you can sometimes find Fremen that were captured and are willing to follow you. At other times, you will find a Harkonnen officer that can you capture, who will tell you about nearby targets in exchange for his life.

In time, you will work your way towards the Harkonnen palace, where the enemies become harder. The final goal is to storm the palace. Therefore, you have to train troops and equip them with different type of weapons in order to prepare for the battle.

The Book of Dune

On the left side of your interface, at the bottom of the screen, you will see a book. If you click it, you can get some background information about Dune, Fremens, Sand Worms and many other things.

In the book, you can find more in-


2

Paul Atreides arrived on Dune with his father, the Duke Leto Atreides, and his mother Jessica.
They settled in the deep desert, in a palace once occupied by their Harkonnen enemies.
Caladan is the Atreides home planet.

Paul Atreides

Figures 13 and 14 - The Book of Dune separated into different topics. Some of the pages include movie cut scenes

formation about the characters and see some cut scenes from the 1984 movie. It's very interesting to see all these scenes from the movie put into this game.

Hints

If you get stuck at some point and don't know what to do, just talk to your comrades or the people in the castle. One of them is normally going to tell you what to do next. Listen carefully and you will figure out what to do next. Visit your mother often and talk to her, she will tell you when your ability to talk to Fremen telepathically has increased. Only after talking to her can you actually reach further with your mind.

Harvesting is rather easy. Whenever you can, use spice harvesters with Ornithopters, even if you have to buy them. Using harvesters makes a huge difference, with 10 or 20 times increase in production as without harvesters. You should always harvest until a field becomes dark brown, and then move on to the next field, especially early in the game.

Make sure you do not forget the spice delivery for the Emperor, so plan your steps so that you are back in time in the palace. A pleased Emperor will only demand a little more spice, and often grants you extra days before asking for a new shipment.

A worm can bring you anywhere,

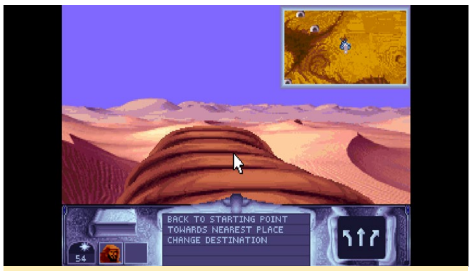


Figure 15 - You can do everything with a worm that you do with a Ornithopter, even searching in the dessert for new sietches

but flying can not get you close the enemy borders. There are also four types of weapons that each Fremen can have in your army: krys knifes, laser guns, weirding modules and atomics. Except for atomics, you can buy everything from villages. In fact, before you begin your first fight, you should already be completely equipped up to weirding modules before you attack your first Harkonnen fortress. This will guarantee you an easy victory.

Learn how to haggle with Smugglers, since they all reduce their prices. Even if they say "forget it," try over and over again since they will always give in, so learn how often you can press them. Some smugglers can be pressed three times, while others can only be pressed twice. They will eventually give you the lower price.

Why I like this game

I have probably played through Dune ten times already. I like the strategy, the troop management, and having those things paired up with a nice adventure where you have to talk to the people at the right time and the right place, while

(continued on page 29)

FRUIT MIDI

BUILDING A GRAPE PIANO

by Georg Mill

One of the things which really creates joy and connects people from all over the world is making music. My father bought me an inexpensive, older drum kit when I was 15 years old. Since then, I became aware that music is one way of getting in touch with people. People love to hear music, and if it is good from their point of view they begin to dance, maybe even with each other.

So if music is connecting people, why shouldn't we connect people to a computer to let them make their own sound and let them feel even better? The problem is that a computer is relatively expensive, and the software for generating sound is even more expensive. The peripherals for human interaction with music software such as keyboards, e-drums or an Ipad is yet another expense. So perhaps these are the reasons that people prefer listening to music instead of making it on their own. My goal was to build an easy to use instrument that is cheap, mobile enough to carry it with you in your pocket, and which produces a lot of fun.

So it should use things that are as readily available as possible for input (in this case: a bunch of grapes, but it could also be strawberries if you like that more). It should be also be able to play 48 different sounds, run on battery power, and have low power requirements.

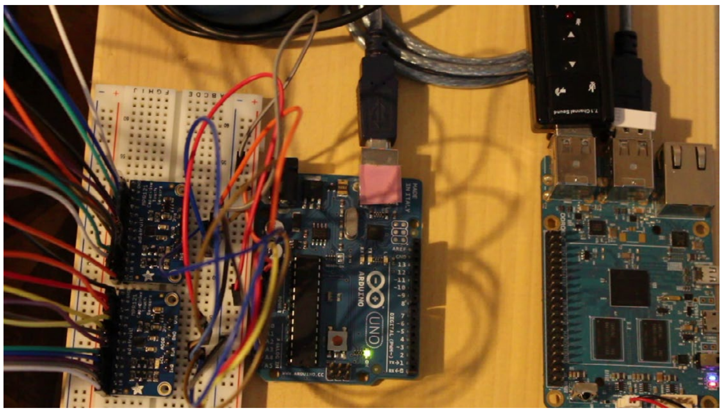
Making Music with the ODROID-C1

At the beginning of this year I got my first ODROID (an ODROID-C1). It is very inexpensive and has all of the requirements for producing high quality music. It is possible to install one of several Linux distributions and immediately have access to high-end, open-source software for pro audio music production. Some of these tools are the Jack Audio Connection kit (http://jackaudio.org), Ardour (http://ardour.org) for recording, editing and mixing, as well as a number of additional professional music production tools. For a brief overview of many of the audio software packages, visit the Unbuntu Studio page at http://bit.ly/1UcARgK.

The ODROID is a perfect platform for musicians, and for those who want to become a musician. It is inexpensive, open-source, and has plenty of GPIO pins available to interface with many types of hardware.

The MPR121 Touch Controller

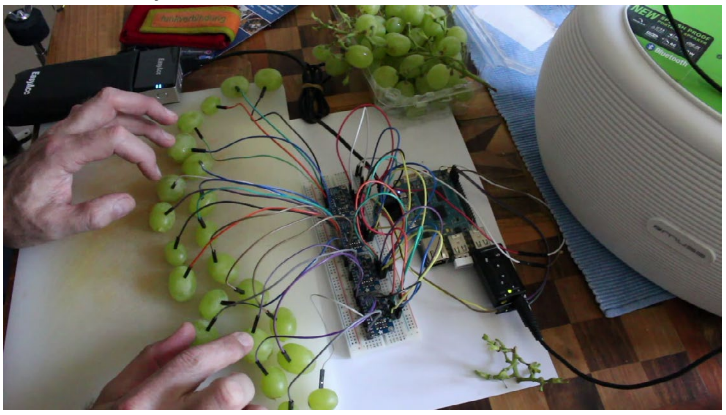
Some time ago, I found a special type of hardware that allows you to literally get "in touch" with your ODROID. This hardware is the MPR121 proximity capacitive touch sensor from Freescale Semiconductor (http://bit.ly/1M1BPeo).



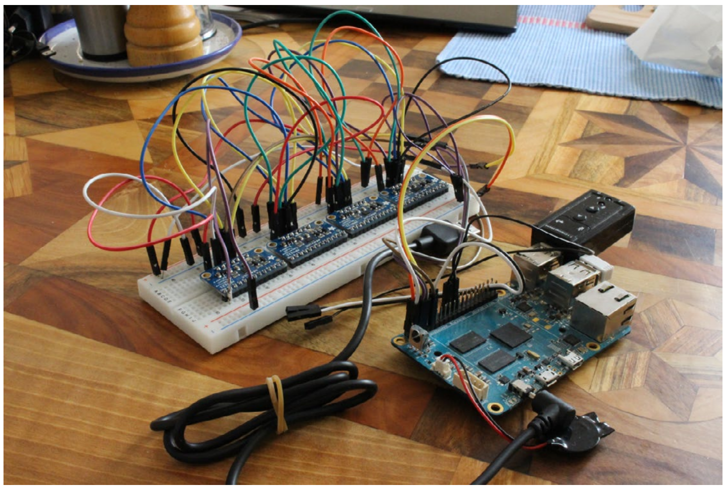
The MPR121 allows you to connect up to 12 electrodes to your single board computer using the I2C bus. It is even possible to connect up to 4 of these controllers to the bus at the same time. To make life easier, a breakout board for these chips is available from Adafruit (http://bit.ly/1MC54Ez), along with a really easy to understand tutorial on assembling and wiring the breakout. Adafruit also provides an MPR121 Library on GitHub (http://bit.ly/1fPiFKR) for using the chip in your projects, whether for music or for other applications.

The Bunch of Grapes MIDI Piano

This was the initial starting point for me in developing a very special kind of MIDI piano, the "Bunch of Grapes MIDI Piano" (http://bit.ly/1MyNfp5). This project provides a piano that allows you to connect just about anything that is conductive, or has conductive capacitance. In this case it is a bunch of grapes, but it could be almost anything that provides this type of electrical characteristics, including metal strips, conductive thread, conductive rubber or tape, flexible PCB, and a number of other things as well (http://bit.ly/1DezERP).



The first time the MPR121 caught my attention was on my last birthday. On that day I received a gift from my parents of an Arduino clone called the "Touch Board", produced by Bare Conductive (http://bit.ly/1zNiHK5). This board has an MPR121 chip and a lot of other nice features on board that allow you to easily start making projects without having to know anything of the technology behind it. This is one of the best places for a beginner to get started with the MPR121. A lot of tutorials can be found at http://bit.ly/1JVOHg5. After poking around, I found the breakout boards from Adafruit mentioned previously, and experimented with them on an Arduino Uno.



MIDI on an Arduino

It is possible to convert an Arduino into a MIDI instrument with additional hardware as described at http://bit.ly/1DfqggB, or as a total software solution. A very nice step-by-step solution for MIDI hardware can be found at http://bit.ly/1IOHyne. If you are not familiar with soldering or this is all too complicated simply make use of an Arduino MIDI shield (http://bit.ly/1hb0PmD).

If you want to keep it as cheap as possible and get results quickly, you can use the software solution for this. It is called "ttymidi" (http://www.varal.org/ttymidi/). Just plug the Arduino into your computer via usb and start ttymidi on your computer with the following command:

```
$ ttymidi -s /dev/ttyACM0 -b 9600 -v
```

This will allow external serial devices to interface with ALSA MIDI applications. This also makes it possible for you to use it with the Jack Audio Connection Kit on your ODROID or any other computer.

Audio and MIDI Latency

The following step is optional, and if you want to keep it simple, just skip this section concerning latency and kernels. At this point we'll dive into a problem that is called latency, and is especially important for live performance audio. My aim was to build a simple to use, battery-driven live MIDI instrument with a high fun factor. You can keep audio latency (http://bit.ly/1SQTW5n) under better control with a low latency kernel (soft real time), or for even tighter control by using with a full preemptive realtime kernel.

Building a full preemptive realtime kernel

The Hardkernel team has made the sources for the ODROID-C1 kernel available on at http://bit.ly/1OT6uJ4. After cloning the repository you have to apply the original rt-kernel patches from http://bit.ly/1MXL4te. If you want to know more about these patches you can find it here: http://bit.ly/1SorwVx. Applying these patches can be difficult, and is not a trivial task for beginners. So if you don't want to dive into it this deeply, just download the kernel sources from my blog http://bit.ly/1RE1Osr and compile it as described at http://bit.ly/1EuxrzR. Now you have an operating system on your ODROID-C1 that is able to let you use your MIDI instruments in real time.

The Jack Audio Connection Kit and MIDI

Jack is a sound server for linux. The Jack Audio Connection Kit provides many more possibilities for connecting your MIDI gear (even over a network), but it is also optional and not necessary if just you want a simple setup. You can read more about these possibilites here: http://bit.ly/1OFqzlz.

If MIDI is new for you you should first read http://bit.ly/1DW25Po which is an excellent intro on how to get MIDI running on your ubuntu computer/ ODROID.

Interfacing the MPR121 to the ODROID-C1

The ODROID-C1 comes with a lot of GPIO pins, so my next step was to connect the MPR121 directly to the ODROID-C1. This would make it even easier to use the chips. This can be done pretty easily because the MPR121 uses the I2C bus which will first need to be activated on the ODROID-C1 as follows:

1. Load the driver:

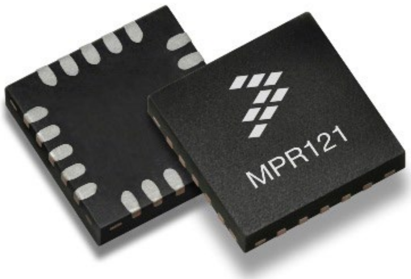
```
$ modprobe aml_i2c
```

2. Start on boot

```
$ sudo echo "aml_i2c" >> /etc/
modules
```

3. Reboot

A very detailed howto can be found at http://bit.ly/1VRxyx6.



This is a lot of work, but when you are all through, you'll be able eat your organic MIDI instrument, and never be hungry again at one of your gigs!

Programming the MPR121 Touch Controller

To get the MPR121 working, I used Python as it is a common language. It's not that I like Python very much, since I come from the Java and C/C++ world, but it is much simpler to use.

Adafruit provides a Python library for the MPR121 (http://bit.ly/1KKhnOU). To get Python running on the ODROID-C1 you can use the ported WiringPi2-python libraries from @mlinuxguy (http://bit.ly/1DjLUR5). The pinmap at http://bit.ly/1JIb9gV is useful to connect the ODROID-C1 and the MPR121.

Generating Sounds on the ODROID-C1

There are several possibilities for generating the sound, including directly using a .wav file stored on the ODROID's file system, or generating a MIDI signal for even greater flexibility. In our case, we'll use MIDI. First, we will first have to install an additional Python library called python-rtmidi (http://bit.ly/1fXpbiT) which is in fact a wrapper for rtmidi (http://bit.ly/1fXpASk).

Rtmidi is a set of C++ classes (RtMidiIn, RtMidiOut and API-specific classes) that provide a common API (Application Programming Interface) for realtime MIDI input/output across Linux (ALSA & JACK), Macintosh OS X (CoreMIDI & JACK), and Windows (Multimedia Library) operating systems. RtMidi significantly simplifies the process of interacting with computer MIDI hardware and software. So if you want to write a program in C++ instead of Python this would be a good choice.

You have to download and compile python-rtmidi. Then, you can generate MIDI notes as shown in the following examples:

```
note_on = [NOTE_ON, 40+i, 112]
Play Note: MIDI Channel 1, Middle
C, Velocity 112
note_off = [NOTE_OFF, 40+i, 0]
Stop Note Play
```

The Adafruit website did not have any examples for using four MPR121 boards working at the same time, so I had to write some code myself. There was a helpful entry on the Adafruit support site (http://bit.ly/1OJ9jvD) that helped me to get it running quickly.

Don't forget to first start your MIDI software and connect it with the right port. Otherwise silence is the only thing you will hear.

MIDI Software Setup

Here are the steps to setup your MIDI software:

1. Fire up qjackctrl and configure it to use the right (USB) Soundcard. See details for this setup at http://bit.ly/1CzBRXC

2. Install the a2jmidi daemon from http://bit.ly/1UgHMFu

a2jmidid is a daemon for exposing legacy ALSA sequencer applications within the JACK MIDI system. It is much easier to mmake MIDI connections with this daemon.

3. Start your sound generator. One open source software synthesizer is zynaddsubfx, and is capable of making countless numbers of different instruments (http://bit.ly/1LZ49fT)

4. Open a console window and type

```
# python touchMidix4.py
```

The script establishes a connection with the following lines:

```
try:
    midiout, port_name = open_
midiport(0, "output",
            api=rtmidi.API_UNIX_
JACK,
              client_
name="sensors", port_name="MIDI
Out")
except (EOFError, KeyboardInter-
rupt):
    sys.exit()
```

To get information about the available ports, and play a testnote, write a Python test script and execute it:

```
import time
import rtmidi

midiout = rtmidi.MidiOut()
available_ports = midiout.get_
ports()

if available_ports:
    midiout.open_port(0)
else:
    midiout.open_virtual_port("My
virtual output")

note_on = [0x90, 60, 112] # chan-
nel 1, middle C, velocity 112
note_off = [0x80, 60, 0]
midiout.send_message(note_on)
time.sleep(0.5)
midiout.send_message(note_off)

del midiout
```

Be sure that the MIDI port (MIDI port 0 In this case) is available on your system and connect it with midiout. open_port(0) in your script.

Other modifications

Four MPR121 breakout boards can connect your ODROID-C1 with a bunch of grapes, bananas, or anything else fun and tasty to allow you make music while keeping you from getting hungry after a stressful gig. Have fun, and check out my videos and Fruit MIDI blog at http://bit.ly/1UgIRgE.

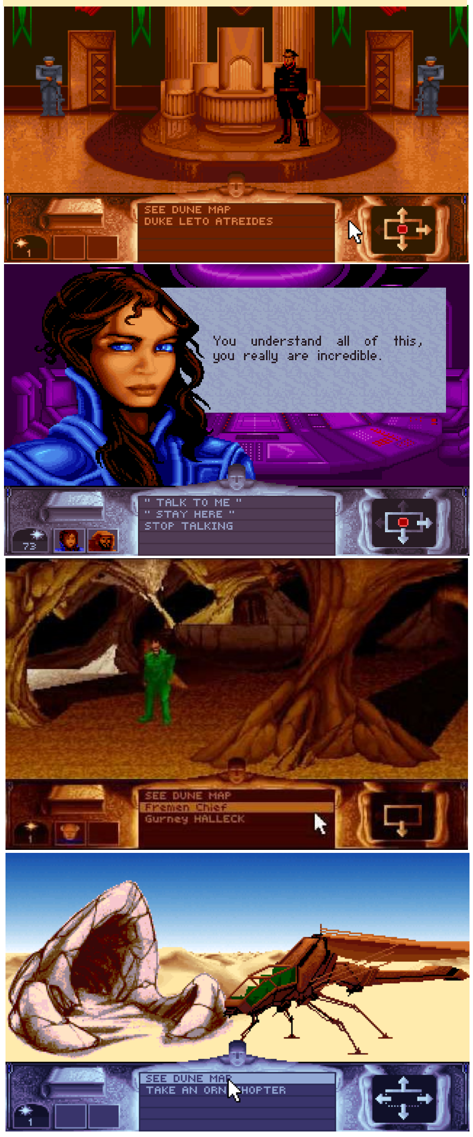
This is all that's left of our grape piano

trying to find hidden things and learning more about the entire universe.

Besides that, the soundtrack is really amazing! It is no wonder that they later created "The Spice Opera" as a standalone production. Together with the complete voice acting in the game with every character and scene speaking to you, it makes the sound really impressive for its time and even today, since many modern games do not have full voice acting for every aspect.

I also really like the drawn graphics, which do not age (in my opinion) and of course, the character progression. Over time your character's eyes turn blue, and you are able to reach further and further with your telepathy. It will always be one of my favorite games and I will probably finish it many more times.

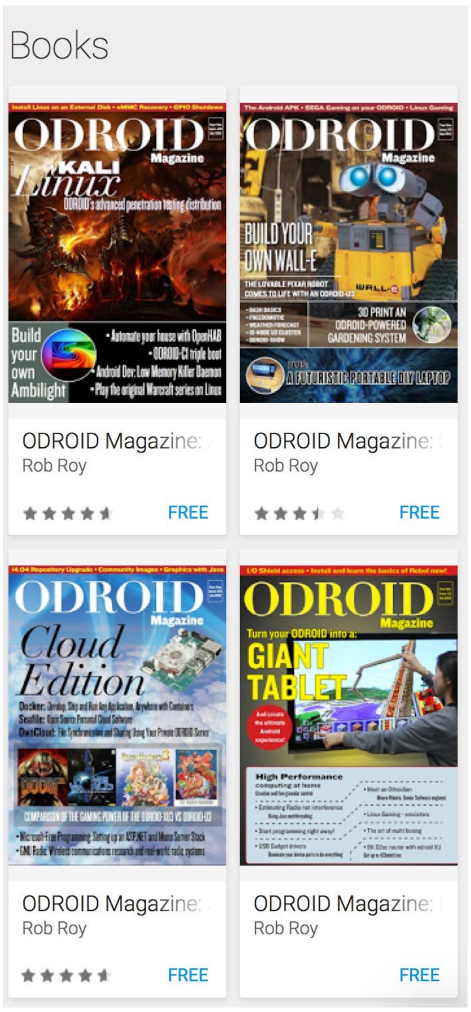
Several screenshots from Dune



ODROID MAGAZINE NOW AVAILABLE ON GOOGLE PLAY STORE

by Rob Roy

ODROID Magazine is now available for download from the Google Play Store and Google Books at http://bit.ly/1IZJELO. Each month, we upload the magazine to the store so that you can have your favorite articles available in one place, all the way back to the first issue from January 2014. Share them with your friends, keep them on your phone, and submit your own review of each issue conveniently. And best of all, its free!



Books

ODROID Magazine: Rob Roy FREE

ODROID Magazine: Rob Roy FREE

ODROID Magazine: Rob Roy FREE

ODROID Magazine: Rob Roy FREE

MEET AN ODROIDIAN
NICOLE SCOTT
MULTI-FACETED ARTIST AND SOCIAL MEDIA GURU

edited by Rob Roy

Please tell us a little about yourself.

I am a one-stop shop Creative Digital Strategist, offering a myriad of services for my clients ranging from print and web design, web migrations and web programming, video production from pre to post, to optimization of online presence and inbound marketing strategies including newsletters, buildout, content curation and management on social media platforms, as well as working with SEO, analytics and Google Adwords. I assist also with training others on how to use digital platforms efficiently, as well as sharing the best tips and tricks to improve their metrics and return on investment (ROI). I have varied interests in both creative and analytical fields and am somewhat of a polymath. I took one of those tests to see whether I was right- or left-brained and the result was centered at 50/50.

How did you get started with computers?

I think the first ones that I used were the Speak and Spell and Speak and Math computers from the early 1980s. Sure, they may have been simply talking calculators, but I really loved those! But as far as desktop units, I began using PC's in high school during the late 1980s and early 1990s. At my dad's land surveying business, I used the PC to print dot matrix banners. I learned bookkeeping on Lotus 123 and backed up the day's work on floppy disks. I also learned computing in high school on the Macintosh II, so I was familiar with both the Windows DOS and Apple operating systems when typewriters were still popular and computers were considered the newest technologies. I was resourceful, and liked exploring all the bells and whistles with all the new software programs coming out on the Macintoshes.

I remember the days when I could boot my Macintosh system from a zip drive whenever my Quadra gave the sad face. I was online as soon as there were 14.4k modems and recall how happy I was just to have those dial up modems giving double, triple, and quadruple speed. When DSL would go down and I'd have to revert back to a 128k modem to be online, I was reminded how slow everything was back then. Advancements in computing technologies in my lifetime has been impressive. I'm amazed at human ingenuity in this regard, having used big tower computers and cathode ray tube monitors, and seeing that hardware evolve and transformed into something even better that can fit in the palm of my hand.

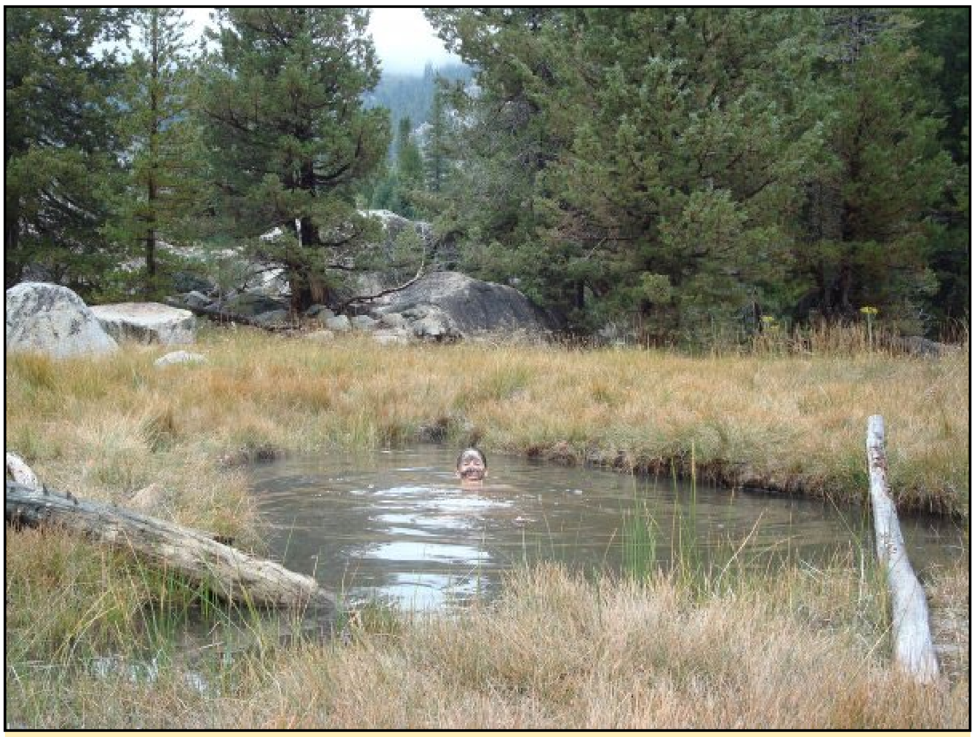
Our Art Editor Nikki, ready to devour a delicious vegan taco!

What attracted you to the ODROID platform?

I heard about micro-computers when the Raspberry Pi was first introduced, but as a front end user, I was attracted to ODROIDs more because of the robust features, additional peripherals, and flexibility of using the Linux or Android OS for a fraction of the cost compared with what I was used to spending on a computer. Having used Macintosh computers for two decades, it always cost near US$2000 to get a new unit. I was thrilled to learn I could do so much on the ODROID devices, and began using primarily an Android OS, because I had already begun using the Samsung Galaxy phone a year before and am familiar with that interface. I first began using ODROIDs during the U2 generation.

How do you use your ODROIDs?

I have different units for tasks ranging from personal computing, such as using Android for games, internet



Nicole enjoying the best that nature has to offer at the hot springs in the John Muir Wilderness - complete with a mud facial, all for the price of a 10 mile backpacking hike!

browsing, mobile apps, and streaming videos. I also use Linux in my ODROID-XU3 as an in-house private server for programming websites offline, which decreases the time spent uploading changes via FTP to a remote server. I can continue to work on my server even when the Internet connection is down, which has happened several times when a squirrel ate through our wires!

Which ODROID is your favorite?

Although the newer models have more powerful capabilities and can be used for many things, I still love the U2 the most. The hardware is the most attractive to me, and I really hope they can make a newer model that reintegrates that same type of design with the integrated metal heat sink. They just look so amazingly cool, and they don't have noisy fans or overly bright lights.

What hobbies and interests do you have apart from computers?

Outside of work, I like playing music as a singer, percussionist and drummer, creating songs on music software, producing animated videos, and studying esoteric subject matter or anything of interest in the moment (lifelong learner). I also enjoy gardening, hiking and backpacking, playing tennis, practicing yoga, watching movies, producing photography and videography, and engaging in intellectual discourse in forums related to subjects ranging from pantheism and philosophy to being solution-oriented in addressing societal problems. I create art with collages or paintings, write poetry, blog, and travel. One of my projects is being involved in the non-profit organization Meiklejohn Civil Liberties Institute, advocating peace law and human rights, and I also aspire to produce my own work as either documentaries or interactive new media. I like cooking vegetarian. vegan meals, and making raw desserts, along with organic and native gardening.

How do you see ODROIDs benefiting future generations?

Since I acquired a number of new ODROID units as a result of being involved with the ODROID magazine as an Art Editor, I would like to provide the units to my children's school for the computer class and get the middle school children excited about ARM computers, and what you can do with this technology. Personally, I see these ODROID units being beneficial insofar as they demand less resources and can operate with such a low wattage that a simple small solar unit is all that's needed to keep them running. Thus, societies and cultures with marginal means in third world countries would be able to purchase these computers far easier than the over-priced Apple models, for instance.

Now that I've been involved in fundraisers for documentaries on sites like IndieGoGo, I'm a considering running a crowd-sourced fundraising campaign to help inner city schools outfit their classrooms with ODROID computers, because you really don't need to spend thousands of dollars per computer to be able to use the Internet for checking your email or playing games, which is what most school-aged kids do. Open-source computing is the way of the future and will bridge the gap between the haves and the have-nots, allowing those with less money to be included rather than excluded, just because they don't have thousands of dollars to buy a computer.

What advice do you have for someone wanting to learn more about social media?

Most anything you want to learn is accessible online, but to streamline the learning process, sometimes it's more valuable to consult with others (like myself) who have extensive experience and knowledge to help you learn the ropes and help you to achieve your goals. It's certainly faster that way, because you can decrease the amount of time that it takes to figure everything out on your own.