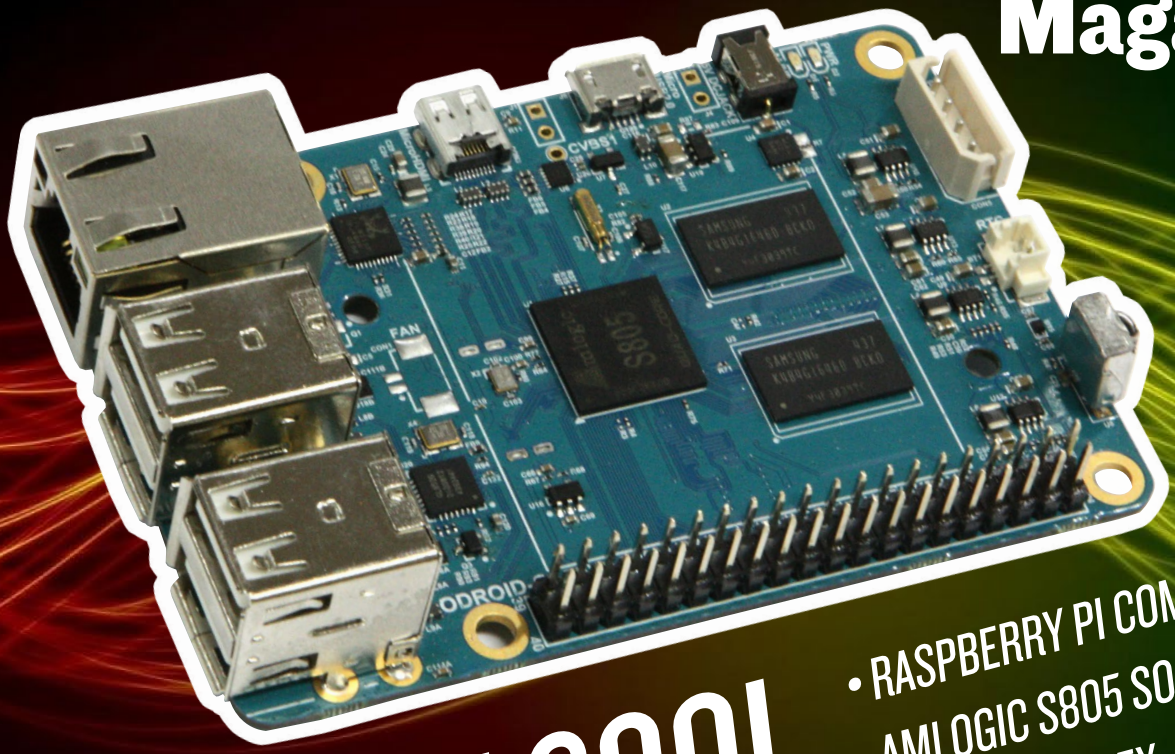
Android Development: Android Init • Mini ZX Spectrum • GameCade

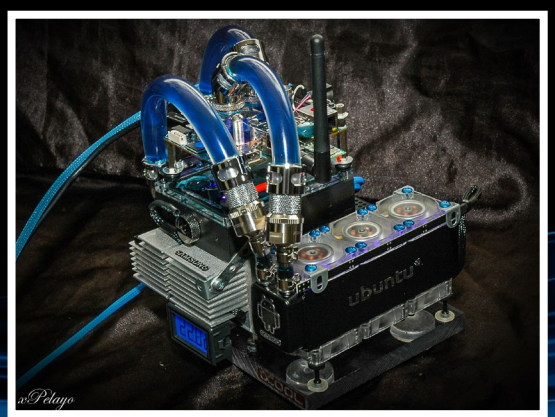
ODROID Magazine

Year One
Issue #12
Dec 2014



GIGABIT COOL ODROID-C1

- RASPBERRY PI COMPATIBLE I/O
- AMLOGIC S805 SOC
- 4 X ARM CORTEX-A5 @ 1.5GHZ
- ARMV7 ARCHITECTURE
- ARM MALI-450 MP2 GPU @ 600MHZ
- 1GB 32BIT DDR3 @ 800MHZ
- 10/100/1000 MBIT/S ETHERNET



A 1080P WATER-COOLED XU-E

A MULTI-ODROID LIQUID-COOLED POWER CLUSTER

- OS SPOTLIGHT: MAX2PLAY
- LINUX GAMING: REMAKES

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

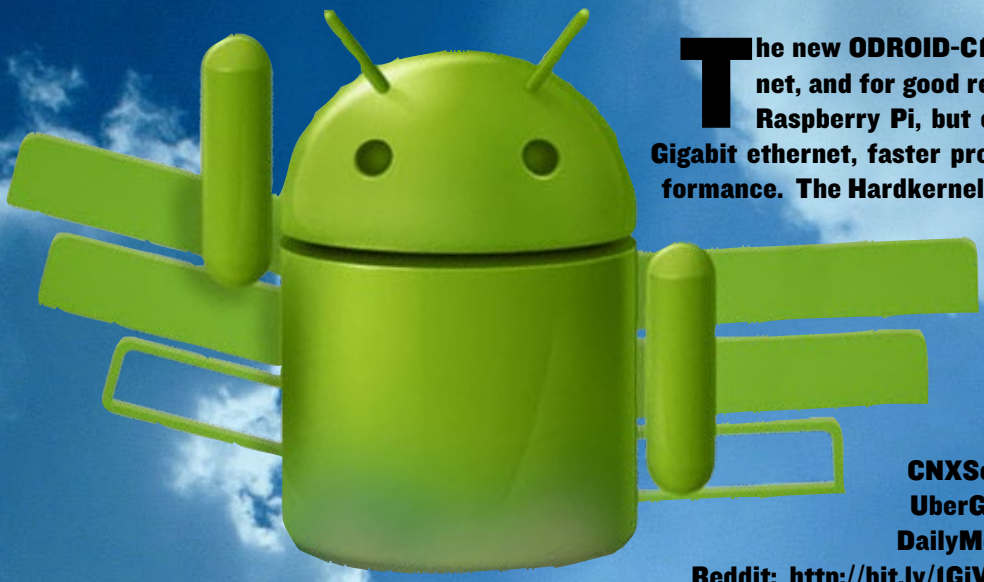
Pollin ELECTRONIC

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

Address: Max-Pollin-Straße 1
85104 Pförring Germany

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

Our ODROID products can be found at
http://bit.ly/1tXPXwe



The new ODROID-C1 has created a huge buzz on the Internet, and for good reason! At $35, it's the same price as a Raspberry Pi, but offers much more in terms of built-in Gigabit ethernet, faster processor speed, and overall better performance. The Hardkernel team has been flooded with orders for the new board, and there have been several high-profile articles and reviews already posted about the advantages of the C1:

Slashdot: http://bit.ly/1wnnj6E
Slashgear: http://bit.ly/1qEJFBc
CNXSoft: http://bit.ly/1ArakI2
UberGizmo: http://bit.ly/1uvAca0
DailyMotion: http://bit.ly/1Arb4qb
Reddit: http://bit.ly/1GjV0KW

If you already have a Raspberry Pi, any of your existing gadgets are directly hardware compatible with the 40-pin I/O interface of the C1. Check out our side-by-side comparison of the ODROID-C1 with the Raspberry Pi B+, as well as a set of computing benchmarks to highlight just how powerful this new board is. If you have any questions about the ODROID-C1, please post them on the ODROID forums at http://forum.odroid.com. The C1 may be ordered from the Hardkernel website at http://bit.ly/1wg54A9.

Hardkernel's more powerful boards, such as the XU3 and XU3 Lite, as well as the versatile U3, have also risen to become the SBC of choice for many mini-computer enthusiasts. A recent survey at LinuxGizmos, in cooperation with the Linux foundation, found that the ODROID-U3 ranked third as a favorite for hardware and software hackers. Its distinct advantage over the first two entries is that it can run Android as well as Linux, which makes it ideal for kiosk projects, Android gaming, as well as app development. The survey results and board comparisons may be viewed at http://bit.ly/1BpMYNz and http://bit.ly/1vJdpbI.

The ODROID-C1 is not the only new hardware available from Hardkernel this month. They've also created a successor to the popular ODROID-SHOW called the ODROID-SHOW2, which includes improvements on the original such as a Li-Po battery circuit and several new LEDs and switches. The price is still $25, and may be purchased at http://bit.ly/1GkIyZS.

Some of our forum members have created amazing liquid cooling systems for their ODROID computers, and the XU-E and cluster systems featured this month are especially impressive. Venkat shows us how to use an RTL-SDR dongle to listen in on FM radio and airplane transmissions, Nanik demonstrates adding boot animations to Android systems, and Tobias continues his popular Linux Gaming series with several DOS remakes that improve upon the original versions. Finally, the coin-operated arcade machine, Amiga 500/2000, and Sinclair ZX Spectrum are given new life by ODROID enthusiasts for replaying their favorite nostalgic games. Bring your quarters!

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
Makers of the ODROID family of quad-core development boards and the world's first ARM big.LITTLE architecture based single board computer.
Join the ODROID community with members from over 135 countries, at http://forum.odroid.com, and explore the new technologies offered by Hardkernel at http://www.hardkernel.com.

HARDKERNEL

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`.

Bo Lechnowsky, Editor

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

Bruno Doiche, Art Editor

Secured his computing necromantic skills after bringing a fiber optics switch back to life, getting his Macintosh back from death, resurrecting a PS3, rescuing his fiancee's T400 with an old-school dd data transplant, and handling the cold innards of his steady job at the data center.

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.

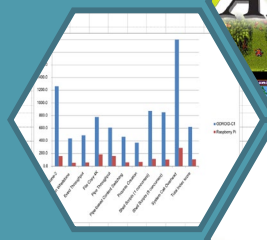
James LeFevour, Art Editor

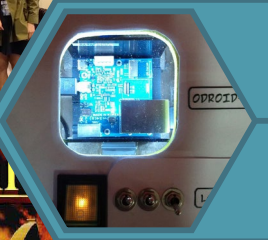
I am a Digital Media Specialist who is also enjoying freelance work in social network marketing and website administration. The more I learn about ODROID capabilities the more excited I am to try new things I'm learning about. Being a transplant to San Diego from the Midwest, I am still quite enamored 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.

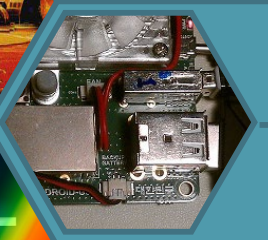
Manuel Adamuz, Spanish Editor

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







LIQUID COOLING

PART I - XU3 CLUSTER

By Uli Abromeit

After finding some small 15x15x5mm heatsinks, I decided to create an ODROID cluster using water cooling in order to reduce its temperature and noise. I started with a single ODROID-XU3 to see if the small heatsinks were powerful enough to distribute heat away from the board. After the initial tests, I connected the cooling system to the rest of the cluster, as shown in the images.

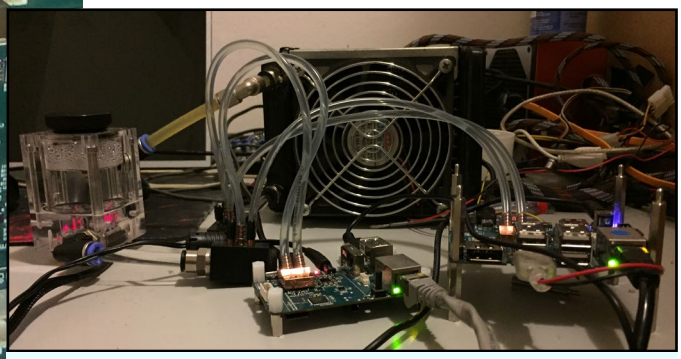
Cooling equipment

Alphacool DC-LT 3600 Ceramic - 12V DC, Alphacool DC-LT Plexi top, and Alphacool reservoir (http://bit.ly/1vDYvJJ)
Alphacool MCX ram copper edition (http://bit.ly/1C3t8Ml)
Alphacool MCX 5x divider (http://bit.ly/1qYh1vr)
Alphacool NeXxus Monsta 140 Radiator with NB-Blacksilent Pro PK2 (http://bit.ly/1Fi5yrA)
120mm radiator
8V pump with reservoir
Adjustable DC-DC step-up convertor to control the speed of the fan and pump

Cluster

10 x Odroid-U3
1 x Odroid-XU
1 x Odroid-XU3
2 x 5V 20A PSU
24 Port NW-Switch
2 x 8 Port-HDMI-Switch

After filling the system, I had some problems with a leaky divider, but now it all runs fine. Using thermal paste instead of a thermal pad reduced the temperature by 5°C, and required custom fittings for the heatsink.



The result of having a super modular portable and resilient computing platform? Ever more ambitious projects!

LIQUID COOLING

PART 2 - XU+E

by @e=MMC2

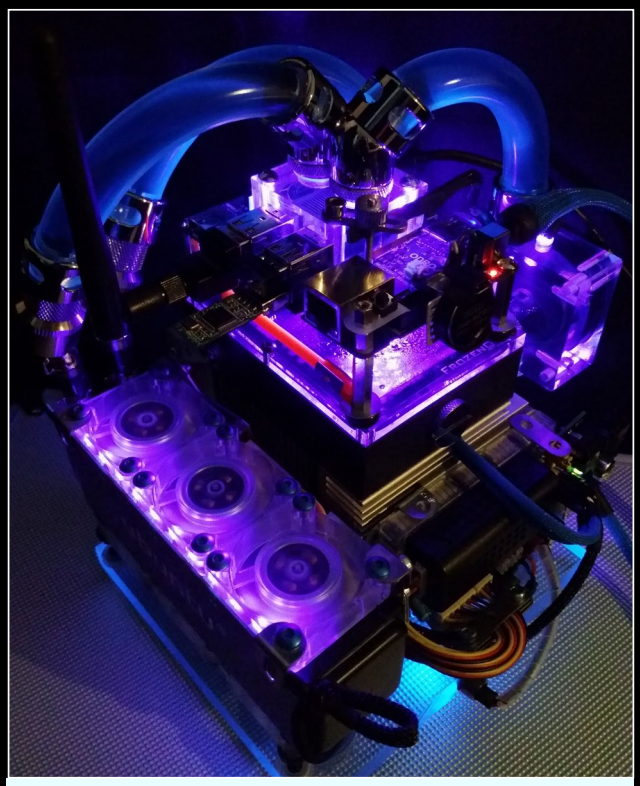
Building a liquid-cooled XU+E took about 5 weeks to complete, with an average of 12-18 hours a day spent working on the project. It weighs a whopping 6.5lbs, with a total cost of around $950, including the board, accessories, cooling hardware, and shipping,

Wiring

Wiring the 18 connections for the 6 fans took almost 2 days to determine how to get the brackets to be perfect. Many hours were spent cutting wires and re-soldering new connectors as well as making them shorter. The power supply proved to be a project in itself. I needed to provide 12v, 5v, 3.3v, and 1.2v to all of the various diff connectors and had to fit 15 connections into the smallest space possible, while still dishing out the various different voltages, along with placing capacitors and resistors.

Breadboard

I found that using a breadboard permanently, rather than soldering up a new PCB, was actually better, since it had a deep base already and can be changed quickly if needed. The main input is a breadboard Arduino power supply, which gave me 5v/3.3v without having to use voltage regulators. I just soldered new leads to the bottom of the input board at the main 12v jack in order to bypass the board and rails, which gave the center of the board the 12v needed for the fans, pump, and LED bars.



WIth the purple lighting, this XU+E looks like a futuristic device that belongs on a spacecraft

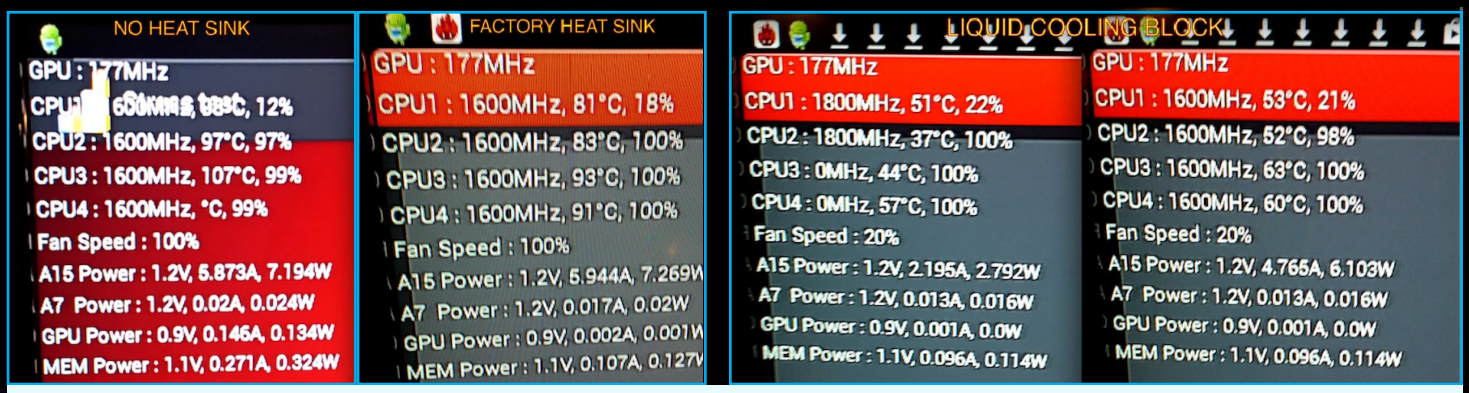
The digital cooling readout and custom OCOOL label are nice touches that make the project look very professional

Fan and pump

I added a fan controller to tone down the fans and lights when I want it to run it silent and dim. The main UV led strip and fans are connected to the fan controller that gears the voltage down to approximately 6v, which keeps the fans just barely spinning but dead silent. I used a polycarbonate bullet-proof plexiglass as the base, then ran a multicolor LED through a hole that I drilled and installed a switch for selecting the color for the base. The pump runs at a constant 12v and seems perfect for the pressure required, so I didn't bother implementing the pump into the fan controller loop. The pump's instructions state that it can run as low as 6v, but I determined that the bottleneck at my radiator could be overcome by pushing the pump at full strength. The pump is quiet and worked out well.

Enhancements

I have ideas for a better setup, but can't really afford to do another one for some time. I think that I can cool the XU-E significantly more, but for now I've shown that the concept works and runs solidly. Although the project could have been built for much less money, I wanted to use



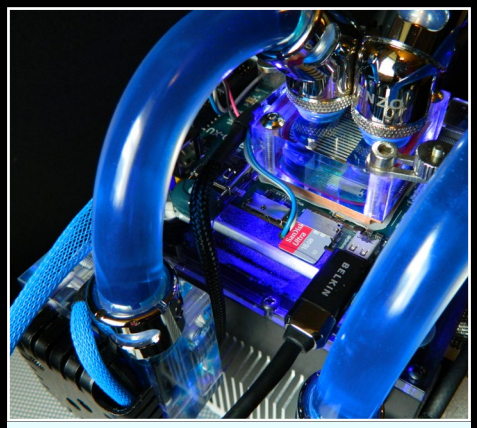
The astounding efficiency of this setup in honest photos of the XU+E temperature measuring screens!

high end parts to give it a more polished look.

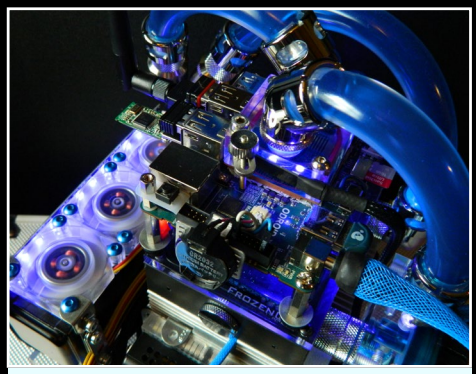
Hardware

Alphacool DC-LT Ceramic 12V DC Pump + Plexi Top
Alphacool NexXxoS XT45 Full Copper Triple 40mm Radiator with 6 fans in a push/pull configuration.
12V fans are 40mmx10mm running at ~6000rpm pushing ~9.5cfm
3/8ID 5/8OD tubing, aside from the 1/4ID to 3/8OD tubing to convert down to the radiator. Only 2 types of 40mm radiator are made, and there are no options for anything except 1/4ID on this breed, so I needed to use a bunch of extra fittings in order to convert it down
Bitspower, Enzotech, and Koolance connectors
Monsoon Free Center compression fittings for the tubing
XSPC LCD temp display with temp sensor for reservoir
FrozenQ Flex tank reservoir
Fesser One UV Blue Non-Conductive Coolant
Fujipoly Extreme Builder Thermal Pad 11.0W/mk
Darkside UV LED strips

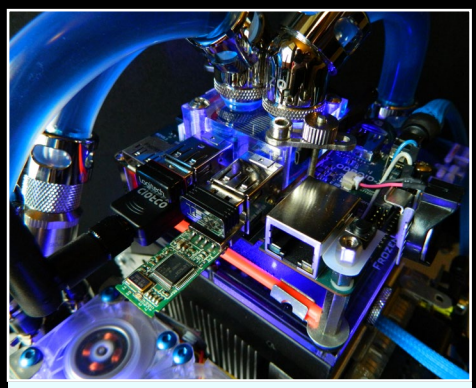
The rest of the parts were o-rings and lighting, as well as various other accessories needed for the power supply setup. The base is an old Macintosh CPU heatsink I found. The rest of the project is mostly held together by an older Erector set that I disassembled. The rubber feet are salvaged from a Playstation controller.



Detail view of the water cooling system



Closeup of the fans used for air cooling

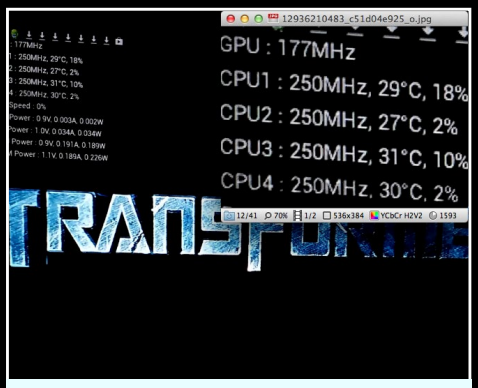


The ethernet and USB ports are still easily accessible

Software

Ubuntu 12.04, 13.10, 14.04 and Server
Xubuntu
Lubuntu
Kali Linux
Debian
Arch
openSuse
Fedora
Suzie
Funtoo
Abacus OS
XBMC 13

Look at that awesome Ubuntu label!



This XU stays cool even when under a lot of stress

OS SPOTLIGHT: MAX2PLAY

EASILY CONTROL YOUR ODROID VIA WEB INTERFACE

Download the Max2Play image in German at http://bit.ly/15b9kYF or English at http://bit.ly/1v07nGY

by Stefan Netzberater

The Max2Play pre-built community image, which runs on both the U2 and U3, features many pre-installed packages for use as an audio and video player, and includes an easy-to-use browser interface to control all of its features. The interface may be accessed from any browser on the local network, and also offers a responsive layout for use on a smartphone via WiFi. If you are new to Linux, you will appreciate the easy setup of features like file sharing, audio player configuration and XBMC updates.

If you are a Linux professional or software developer, you will like the ability to create your own plugins for the Max2Play web interface in order to have easy access to your ODROID. This makes it a perfect solution for headless devices without a monitor or keyboard/mouse connected.

I created the Max2Play image for the ODROID-U2 in the summer of 2013 with a focus on building a standalone multi-room audio and video player. It took some time to finish the first release of Max2Play, because of some tearing and resolution issues with XBMC that have since been resolved. I currently use 2 ODROIDs in my house as a video player and Squeezebox client, and am very happy with the performance.

The most recent Max2Play image is based on the Ubuntu 14.04 Trusty Dev Center image from Marian Mihailescu (@memeka on the ODROID forums), and includes a lot of additional packages and settings, enhanced with services that focus on its usage as media player. The pre-installed system requires minimal configuration by the user to get started.

M2P is a great Ubuntu derivative distribution, which may be a great fit for those who want to have a remote controlled media center

The main goal of Max2Play is to transfer the control of the ODROID to your smartphone or tablet so that a keyboard or mouse is no longer required. The local browser interface of Max2Play, along with mobile apps like iPeng or XBMC Remote, make it easy to use the ODROID as a remote media player.

Browser interface

The browser interface of Max2Play is intended as a community project that can be used by other programmers to enhance their own projects. It is still a work in progress, and the source code is available at http://bit.ly/1pdAfvN.

Configuration and control

Enter http://max2play in the browser of any computer on your home network, or on the ODROID itself, to access the web interface. Use the basic settings on the "Settings / Reboot" menu to adjust plugin configurations and resize the file system. The other tabs are self-explanatory. Further video tutorials, features and instructions, such as how to set up file sharing and how to create plugins, are available at http://www.max2play.com.

The root password for the image is "max2play", and there is also a standard user already created named "odroid" with a password of "odroid".

A hassle free smooth XBMC comes standard and fully optimized on M2P, so fire your XBMC remote on your cellphone and enjoy!

Features and applications

- Multi-language browser interface for configuration. The default configuration web address is http://max2play, with German, English and Italian support
- Audio players include Squeezelite, Shairport, Squeezeslave, Equalizer with Alsaequal for Squeezeslave and Shairport, with custom start parameters for each player
- Squeezebox server can be installed and configured with the Max2Play web interface, with different versions available that are easy to update
- XBMC 13.2 (Gotham)
- File system shares provided by NAS or Samba
- WiFi configuration
- Easy single-click updates in the web interface to reset, reboot, edit the default player, resize the file system, switch languages or timezone, and adjust the keyboard layout
- Manage and install Jivelite as a Squeezeplayer visualization
- Use your device as a call blocker using a Max2Play plugin with blacklists and whitelists, which is easily integrated with Fritzbox
- Activate or deactivate plugins, and adapt the navigation bar for your preferred services
- Advanced settings, such as disabling the blinking of the blue LED of the ODROID, and automatically mounting the USB storage with USB idle if there is no activity
- Pre-configured Asound setup for ALSA, which is used by Audio Player, and Pulseaudio, which is used by XBMC. The audio players are capable of playing simultaneous streams, and supports a software equalizer
- Whenever XBMC launches, all audio players automatically stop
- XBMC 13 features very good video performance (1080p) with no tearing or frame drops, fast interface control and easy integration of 5.1 USB sound cards with digital out, and audio passthrough to an AV receiver (X-FI or LogiLink 7.1)
- Airplay for music, pictures and video streaming directly from a browser. There are still issues to be solved when using video from a camera
- Plugin configuration to add new features and customize navigation
- Easy to use even for non-programmers, allowing configuration of mount points, Samba shares, services and applications
- Support for mobile devices and tablets, so you can control Max2Play remotely from a couch or office

Squeeze light, check! Shairport, check! Squeezeslave, check! Graphic equalizer, check! M2P is a play-it-all solution

For use as a media player, the Max2Play image comes ready to install the latest Squeezebox Server, which is a multi-room audio application with lots of features, using the ODROID as a software Squeezebox. You may install and use things like an equalizer and crossfeed/lowpassfilter, cast media with Airplay using the shairport package, as well as running Squeezelite. You can also mount network shares, connect a USB storage device, and share its content with other network players using Samba or miniDLNA.

Image burner

With the adjustable power button settings using long and short press actions, you can set your own scripts to start when the button is pressed. With this feature, you may use the ODROID to burn images onto SD cards whenever the button is pressed, then signal the success using the blue LED.

Call blocker

The tellows plugin brings call blocker functionality to the ODROID. If you have a router that can handle IP phones, you can easily install the call blocker service and connect it to your router. You can then use your own black and whitelists and the tellows lists in order to block spam calls. If you want to use just the Max2Play web interface, you can install it separately on your existing image from the Github source mentioned above.

File Structure

The Apache web server files for the web interface are located in /var/www/max2play. All of the configuration files, scripts and cache storage are located in /opt/max2play. Actions that require sudo rights use the file "/opt/max2play/dynamicscript.sh" to run. The web interface is written in jQuery and PHP in order to render the HTML

Easy and straightforward configurations are a guarantee that you will take much more time enjoying your ODROID than tweaking it

pages. The language of the pages is adapted to your browser language if available, with English as the default.

Advanced Usage

To create your own plugins for the web interface, you can use the script in /opt/max2play/createplugin.sh to create the correct folder structure and files in /var/www/max2play/application/plugins/[plugin name]. After running the script, the plugin may be activated on the "Settings / Reboot" tab. Refer to some of the existing plugins for examples of writing shell scripts for your custom plugins.

Software notes

There is a known bug that when using the Alsaequal equalizer with Squeezeslave, there is no output through the headphone jack, but the HDMI audio stream does work properly. If you don't need the equalizer, stick to Squeezelite as a player. If there is a way to get it working with Alsaequal, please create a post on the ODROID forums at http://forum.odroid.com.

More information

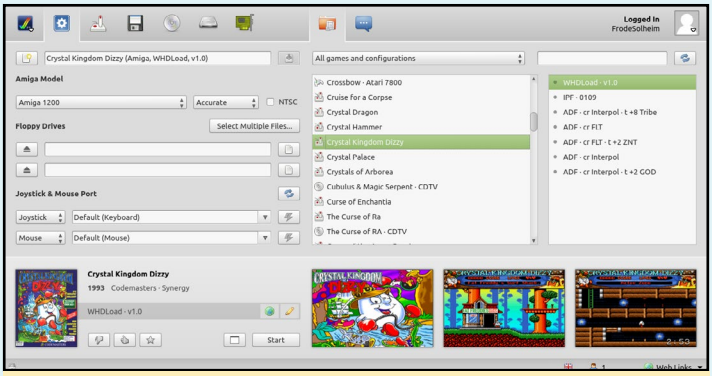
Questions, suggestions, improvements or comments regarding the Max2Play image may be posted on the original forum announcement at http://bit.ly/1te1Edx. The Max2Play website, which is available in German and English, offers pre-built Max2Play U3 hardware packages as well as offering free downloads for those who already own a U3 at http://www.max2play.com.

FS-UAE AMIGA EMULATOR

HAS ANYBODY TALKED ABOUT THE COOLEST COMPUTER BEFORE ODROIDS? WE SURE LIKE TO!

by Tobias Schaaf

The Commodore Amiga was a popular gaming computer in the late 1980s and 1990s, with many unique and fun games available. Although the Amiga is no longer available, the Linux application called FS-UAE, located at http://fs-uae.net, provides an emulation platform for running any Amiga game, including those written for the A1200.



FS-UAE makes the ODROID Magazine team nostalgic for games

To install FS-UAE, type the following into Terminal:

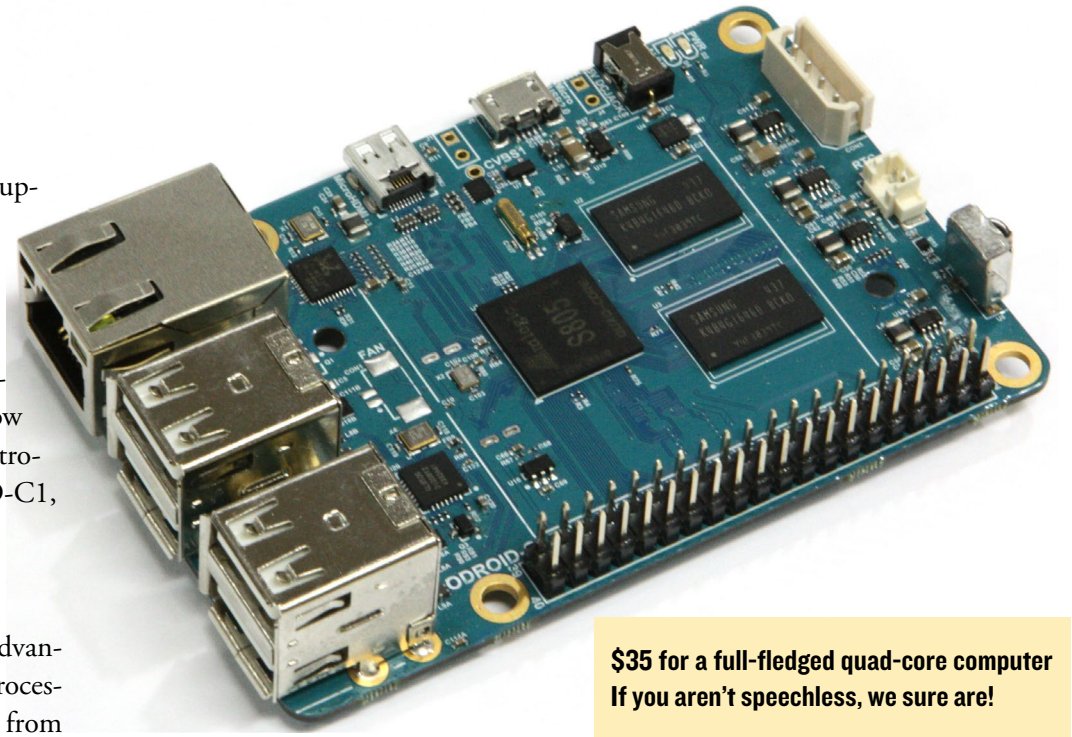
```
$ sudo -s
$ wget http://oph.mdrjr.net/meveric/sources.lists/\
meveric-all-main.list
$ wget -O- http://oph.mdrjr.net/meveric/meveric.asc \
| apt-key add -
$ apt-get update && apt-get install fs-uae \
fs-uae-launcher
```

The Amiga was one of the best 1980s gaming machines available

INTRODUCING THE ODROID-C1

A POWERFUL RASPBERRY PI REPLACEMENT

by Ruppi Kim and Kevin Kim



$35 for a full-fledged quad-core computer
If you aren't speechless, we sure are!

Have you been waiting to upgrade your Raspberry Pi computer, or are looking for a second computer for family, work or college use? Hardkernel's family of ultra-low-cost, powerful ARM devices is now even more affordable with the introduction of the powerful ODROID-C1, available for only $35.

Features

The ODROID-C1 has many advantages over the Raspberry Pi. The processor is an S805 1.5GHz Quad-core from Amlogic with 1GByte DDR3 RAM, Gigabit Ethernet and IR-receiver. The size of this computer is still only 85 x 56 mm with a weight of 40g, and offers silent operation, 2~3W average power usage, and instant portability, since it fits in a shirt pocket.

One powerful feature of the ODROID-C1 is the row of GPIO (general purpose input/output) pins along the edge of the device. These pins are a physical interface between the board and the outside world. The 40-pin interface header includes SPI, I2C, UART, ADC and GPIO function.

An SD 3.01 standard compatible UHS-1 Micro-SD card, as well as the faster eMMC module, can be ordered with the ODROID-C1, and arrives with the popular Ubuntu operating system already installed. Insert the SD card into the slot, connect a monitor, a keyboard, a mouse, Ethernet and power cable, and that's all you need to do to use the ODROID-C1! Browse the web, play games, run office programs, edit photos, develop software, and watch videos right away.

Work and play

The ODROID-C1 is suitable for anyone from professional software engineers building Kinect-drive robotics projects to kids learning to program with the Scratch language. Run the latest Ubuntu 14.04 or Android KitKat operating systems for programming, learning, gaming, media center, web server, office/college work, hardware IO platform and many other applications. The powerful 1.5Ghz quad-core processor, low cost, energy efficient features and massive software library make the ODROID-C1 the perfect modern computer for work or play.

A Tour of the Board

Let's start with a quick tour of what you're looking at when you take it out of the box. It's similar to your typical PC with various added features.

A. Processor

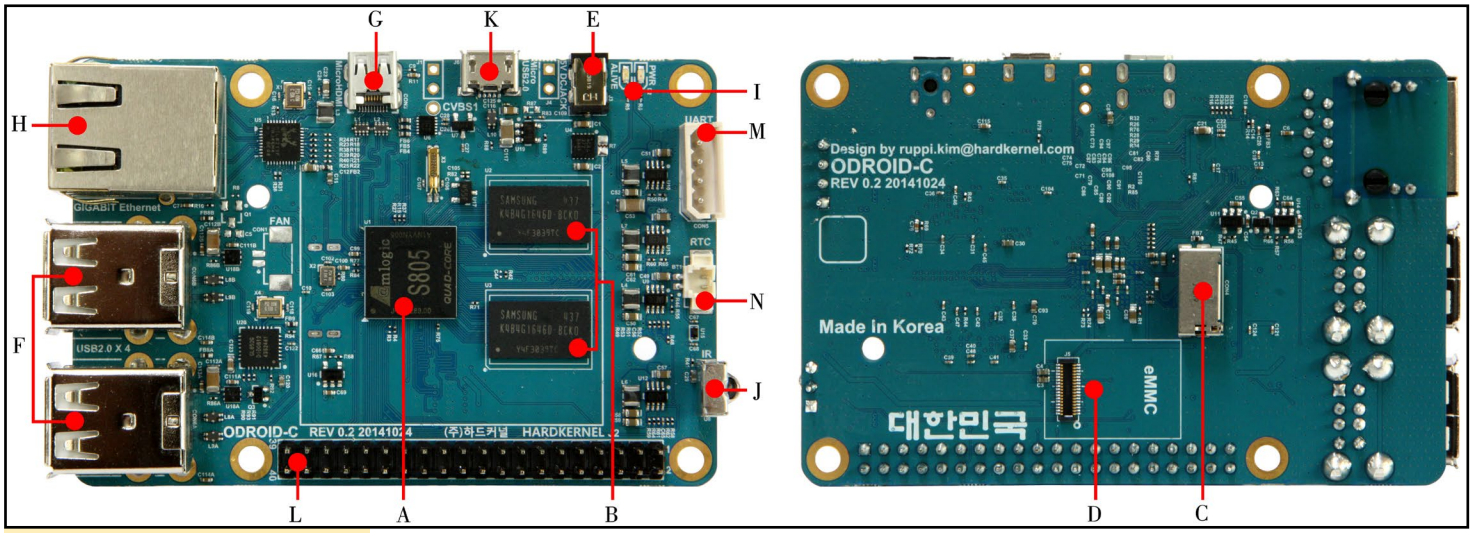
At the heart of the ODROID-C1 contains Amlogic S805 system on a chip, which is built on the quad-core ARM Cortex-A5 ARMv7 architecture and quad-core Mali-450 MP2 GPU.

B. Memory (RAM)

The 1GB memory contains 2pcs of 512MB DDR3 SDRAM. The 800Mhz clocking delivers 1600Mhz of 32bit data traffic via DDR technology.

C. Micro Secure Digital (MicroSD) Card slot

There are two different methods of storage for the operating system. One is by using a MicroSD Card and another is using an eMMC module, which is normally used for external storage for smartphones and digital cameras. The ODROID-C1 can utilize the newer



The ODROID-C1 board in detail

UHS-1 model, which is about 2 times faster than a normal class 10 card.

D. eMMC Module socket

The eMMC storage access time is 2-3 times faster than the SD card. There are 3 size options: 8GB, 16GB and 64GB. Using an eMMC module will increase speed and responsiveness, similar to the way in which upgrading to a Solid State Drive (SSD) in a typical PC also improves performance over a mechanical hard drive (HDD).

E. Power jack

This is for 5V power input, with an inner diameter of 0.8mm, and an outer diameter of 2.5mm. The ODROID-C1 consumes less than 0.5A in most cases, but it can climb to 2A if many passive USB peripherals are attached directly to the main board.

F. USB host port

There are four USB 2.0 host ports. You can plug a keyboard, mouse, WiFi adapter, storage and many other devices into these ports. You can also charge your smartphone with it! If you need more than 4 ports, you can use a powered external USB hub to reduce the power load on the main device.

G. HDMI connector

To minimize the size of the board, we used the Type-D micro-HDMI connector.

ODROID-C1 40pin Layout

	Power Pin
	Special Function
	GPIO/Special Function

WiringPi GPIO#	Export GPIO#	ODROID-C PIN		Label	HEADER		Label	ODROID-C PIN		Export GPIO#	WiringPi GPIO#
				3V3	1	2	5V0				
8		GPIODV.BIT24	I2CA_SDA	SDA1	3	4	5V0				
9		GPIODV.BIT25	I2CA_SCL	SCL1	5	6	GND				
7	83		GPIOY.BIT3	#83	7	8	TXD1	GPIOX.BIT16	TXD_B	113	15
				GND	9	10	RXD1	GPIOX.BIT17	RXD_B	114	16
0	88		GPIOY.BIT8	#88	11	12	#87	GPIOY.BIT7		87	1
2	116	CTS_B	GPIOX.BIT19	#116	13	14	GND				
3	115	RTS_B	GPIOX.BIT18	#115	15	16	#104	GPIOX.BIT7		104	4
				3V3	17	18	#102	GPIOX.BIT5		102	5
12	107	MOSI	GPIOX.BIT10	MOSI	19	20	GND				
13	106	MISO	GPIOX.BIT9	MISO	21	22	#103	GPIOX.BIT6		103	6
14	105	SCLK	GPIOX.BIT8	SCLK	23	24	CE0	GPIOX.BIT20	CE0	117	10
				GND	25	26	#118	GPIOX.BIT21		118	11
	76		I2CB_SDA	SDA2	27	28	SCL2	I2CB_SCL		77	
21	101		GPIOX.BIT4	#101	29	30	GND				
22	100		GPIOX.BIT3	#100	31	32	#99	GPIOX.BIT2		99	26
23	108		GPIOX.BIT11	#108	33	34	GND				
24	97		GPIOX.BIT0	#97	35	36	#98	GPIOX.BIT1		98	27
			ADC.AIN1	AIN1	37	38	1V8	1V8			
				GND	39	40	AIN0	ADC.AIN0			

The ODROID-C1 40-pin GPIO layout

H. Ethernet port

The standard RJ45 Ethernet port for LAN connection supports 10/100/1000Mbps speed.

I. Status LEDs

The ODROID-C1 has four indicator LEDs that provide visual feedback:

The ODROID-C1 status LEDs

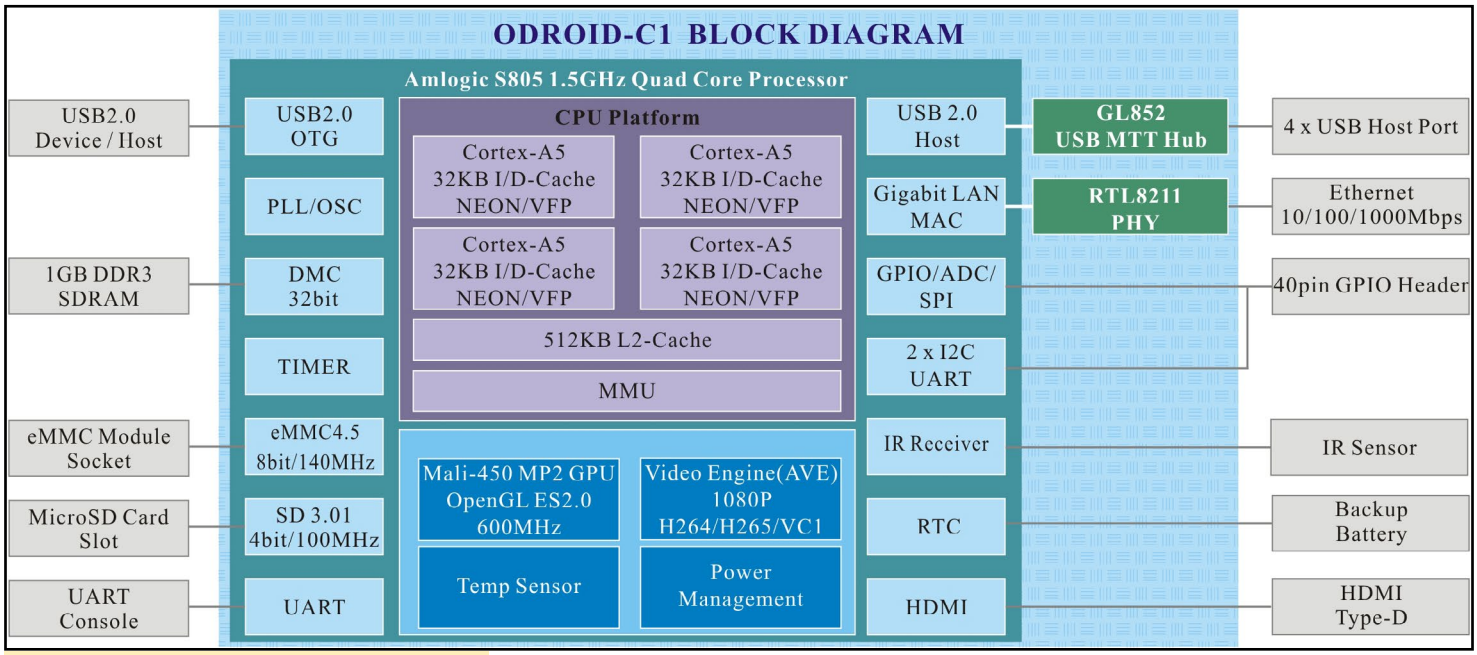
PWR	Red	Hooked up to 5Vpower
ALIVE	Blue	Solid light : u-boot is running Flashing : Kernel is running (heart beat)
ETHERNET	Green	Flashes when there is 100Mbps connectivity
	Yellow	Flashes when there is 1000Mbps connectivity

J. Infrared (IR) receiver

This is a remote control receiver module that can accept standard 37.9Khz carrier frequency based wireless data.

K. Micro-USB connector

You can use the standard micro-USB connector with Linux Gadget drivers on your host PC, which means that the resources in the ODROID-C1 can be



A block diagram showing the architecture of the ODROID-C1

shared with typical PCs. You can also add a micro-USB to HOST connector if you need an additional USB host port. Note that this port cannot be used for power input.

L. General Purpose Input and Output (GPIO) ports

These 40-pin GPIO port can be used as GPIO/I2C/SPI/UART/ADC for electronics and robotics.

M. Serial terminal port

Connecting to a PC gives access to the Linux console. You can see the log of the boot, or to log in to the C1 to change the video or network settings. Note that this serial UART uses a 3.3 volt interface. We recommend the USB-UART module kit from Hardkernel.

N. RTC (Real Time Clock) backup battery connector

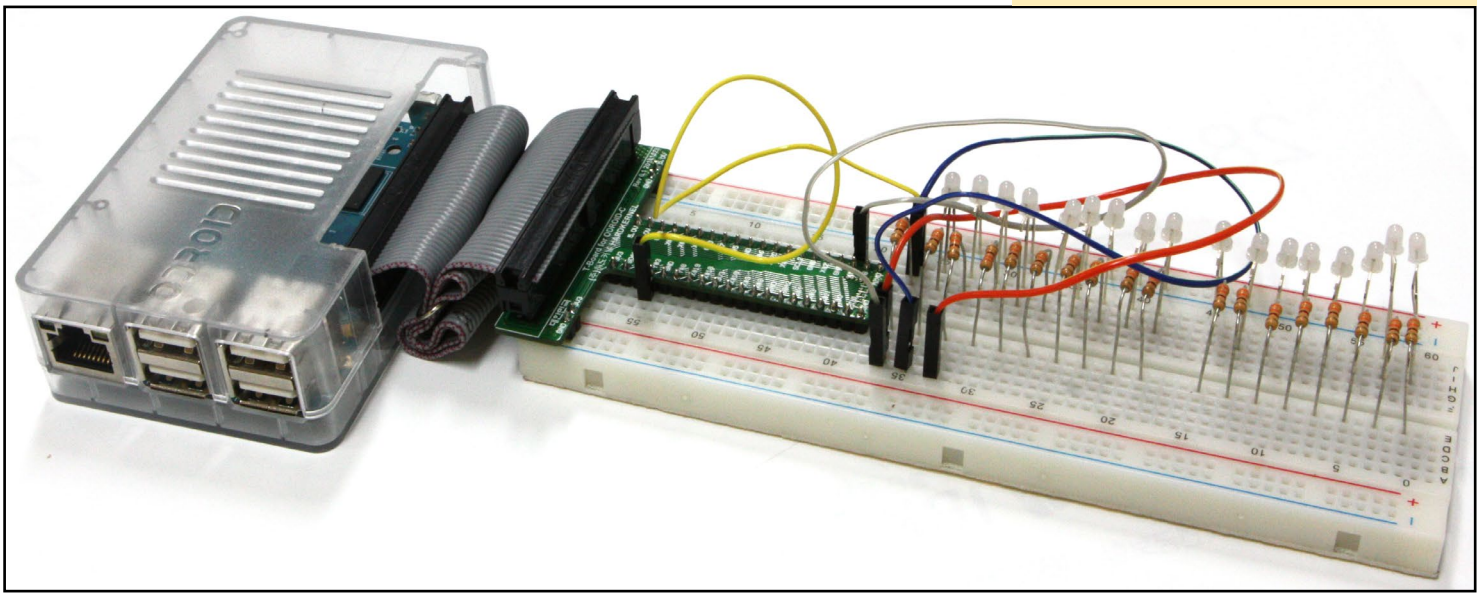
If you want to add a RTC functions for logging or keeping time when offline, just connect a backup battery. All of the RTC circuits are included on the ODROID-C1 by default.

The 40 GPIO pins on an ODROID-C1 are a great way to interface with physical devices like buttons and LEDs using a lightweight Linux controller. If you're a C/C++ or Python developer, there's a useful library called WiringPi that handles interfacing with the pins. We've already ported the WiringPi v2 library to the ODROID-C1.

Please note that pins #37, 38 and 40 are not compatible with the Raspberry Pi B+ 40-pin header. Those pins are dedicated to analog input functions.

To see the ODROID-C1 in action, visit YouTube at http://bit.ly/1wFDwrg and http://bit.ly/1Grw2Hq.

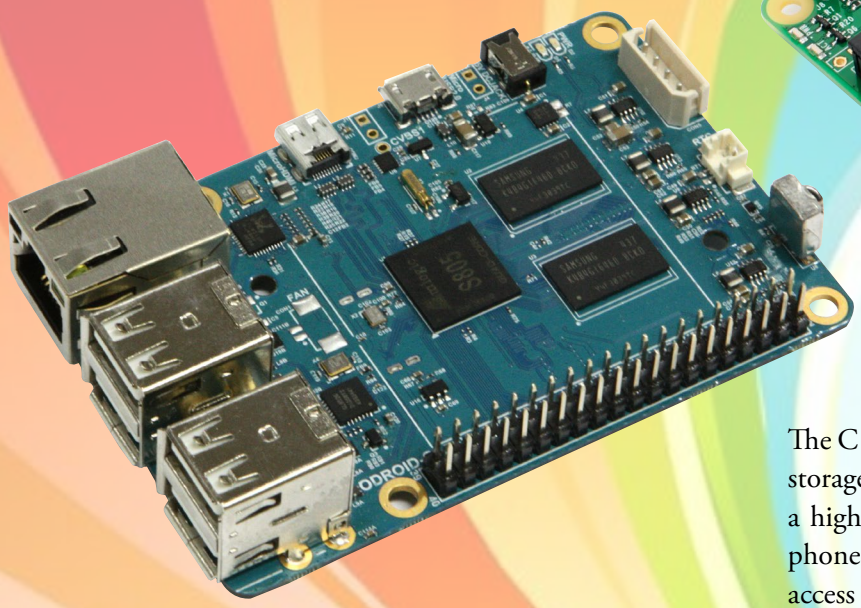
ODROID-C1 prototyping example

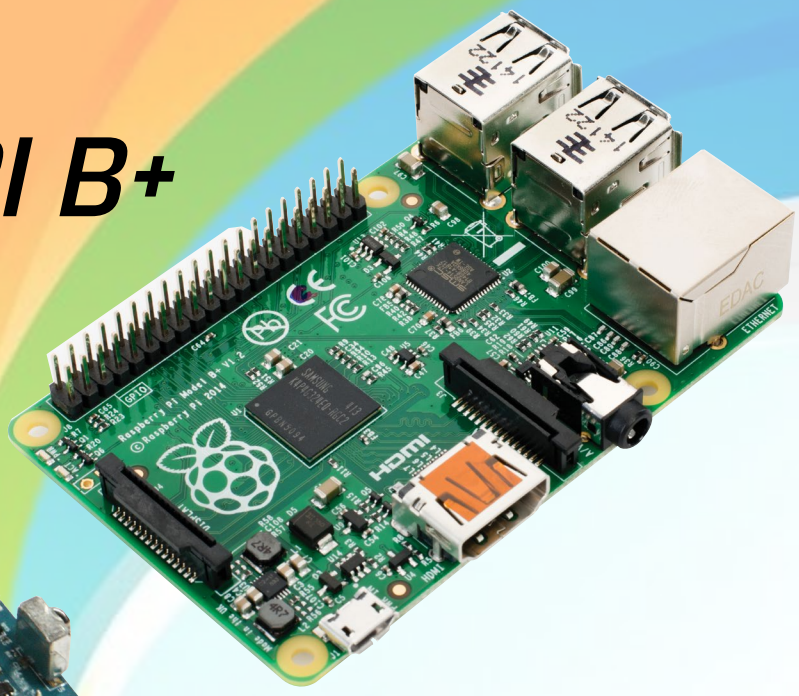


ODROID-C1 VS RASPBERRY PI B+

OUR SECOND AFFORDABLE COMPUTER SHOWDOWN

by Ruppi Kim





The ODROID-C1 is a new Raspberry Pi B+ clone that eclipses the original in terms of performance

Both the ODROID-C1 and the Raspberry Pi B+ are Linux-friendly, cost-effective ARM single-board computers suitable for various applications and purposes. Although the ODROID-C1 is an inexpensive ARM single-board computer, it offers a quad-core ARMv7 CPU, and includes an Amlogic S805 quad-core ARM Cortex-A5 1.5GHz SoC, Mali-450 MP2 GPU, four USB 2.0 host ports, one 10/100/1000MBit Ethernet port, a 1080p video output via micro-HDMI, 1GB of DDR3 system memory, eMMC and micro-SD sockets for solid state storage, and support for Debian Wheezy, Ubuntu Linux and Android operating systems.

Hardware comparison

Compared to the Raspberry Pi (RPi), the C1 has 4 times the number of CPU cores, and the operating clock frequency is about 2 times faster. In addition, the RAM size is also 2 times larger and the RAM access frequency is twice as fast. The C1 also includes a Gigabit Ethernet port that allows higher throughput speeds of around 500Mbps in the real world. The C1 has 4 USB-host ports, as well as a USB-OTG port for fast connectivity with Linux gadgets.

The C1, unlike the RPi, supports a UHS-1 compatible SD Host controller which allows 2 times faster I/O performance. The C1 goes one step further and adds an option to use eMMC storage. eMMC, which is sold separately as an accessory, is a high speed MLC flash memory like those used on modern phones and tablets, and can be removed or replaced easily. The access speed of an eMMC module is typically 2-3 times faster than SD cards. As a bonus, the built-in Real-Time Clock (RTC), Infrared (IR) receiver and Analog-To-Digital Converter (ADC) features on the ODROID-C1 offer many options for building great DIY projects.

Feature comparison of computing performance

	ODROID-C1	Raspberry Pi (Model-B+/512MB)
CPU	Amlogic S805 SoC 4 x ARM Cortex-A5 @1.5Ghz ARMv7 Architecture @28nm wafer	Broadcom BCM2835 1 x ARM11 @700Mhz ARMv6 Architecture @40nm wafer
GPU	ARM Mali-450 MP2	1 x VideoCore IV @ 250 MHz
RAM	1GB 32bit DDR3 @800Mhz	512MB 32bit LP-DDR2 @400Mhz
Flash Storage	Micro-SD UHS-1@100Mhz/SDR50 eMMC storage option	Micro-SD @50Mhz/SDR25 No eMMC storage option
USB 2.0 Host	4 Ports	4 Ports
USB 2.0 Device/OTG	1 Port for Linux USB Gadget driver	Not Available
Ethernet/LAN	10/100/1000 Mbit/s	10/100 Mbit/s
Video output	HDMI	HDMI / Composite RCA
Audio output	HDMI	HDMI / 3.5mm Jack
Camera Input	USB 720p	MIPI CSI 1080p
Real Time Clock	YES (On-board RTC)	No(unless using an add-on module)
I/O Expansion	40pin port (GPIO/UART/SPI/I2C/ADC)	40pin port (GPIO/UART/SPI/I2C)
ADC	10bit SAR 2 channels	No (unless using an add-on module)
SIZE	85 x 56mm (3.35" x 2.2")	85 x 56mm (3.35" x 2.2")
WEIGHT	40g (1.41 oz)	42g (1.48 oz)
Price	$35	$35

Benchmarks

We ran a simple, popular benchmark called Unix-Bench (version 5.1.3) to compare the performance of the two boards. Tests were done using several manufacturer-provided images based on a clean install, and the "apt-get update && apt-get upgrade" commands were first run to ensure that both boards was up-to-date.

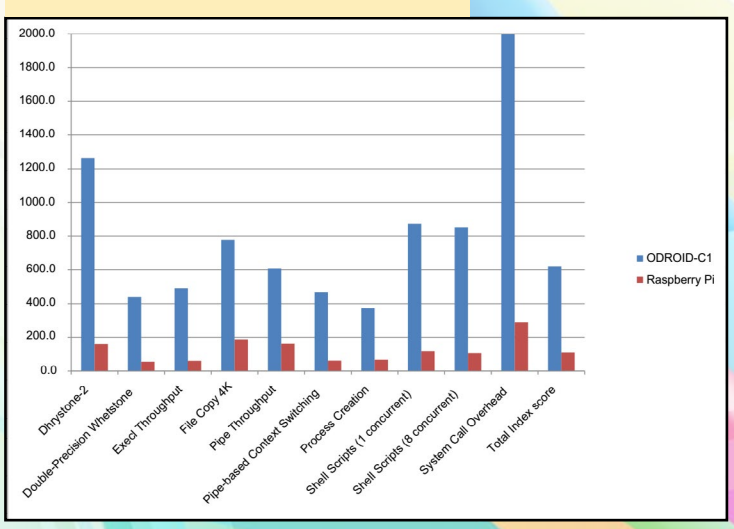
The RPi was clocked at 800Mhz using a Sandisk UHS-1 8GB SDCard running the Debian Wheezy OS. The C1 was clocked at 1.5Ghz using an 16GB eMMC with Ubuntu 14.04 OS. Both units were powered by a 5V/2A power supply and connected to the 1920x1080 HDMI output. Note that in order to utilize all four cores in the C1, the "./run –c 4" command was used.

Benchmarks (Index Score)	Raspberry Pi	ODROID-C1	Ratio
Dhrystone-2 using register variables	162.1	1262.8	7.8
Double-Precision Whetstone	56.2	439.6	7.8
Execl Throughput	61.6	489.4	7.9
File Copy 4096 bufsize 8000 maxblocks	187.9	778.4	4.1
Pipe Throughput	164.1	610.4	3.7
Pipe-based Context Switching	62.7	467.0	7.4
Process Creation	68.2	371.8	5.5
Shell Scripts (1 concurrent)	117.2	874.4	7.5
Shell Scripts (8 concurrent)	106.2	853.8	8.0
System Call Overhead	290.5	1999.7	6.9
Total Index score	109.8	622.3	5.7

Benchmark results of CI vs Rpi

The results show that the Dhrystone-2 benchmark is about 8 times faster on the C1. File I/O benchmark is about 4 times faster because of the faster storage speed of the eMMC module. Overall performance test result show that the C1 is approximately 6 times faster than the RPi, even though the price of C1 is exactly same.

Performance comparison of CI vs Rpi



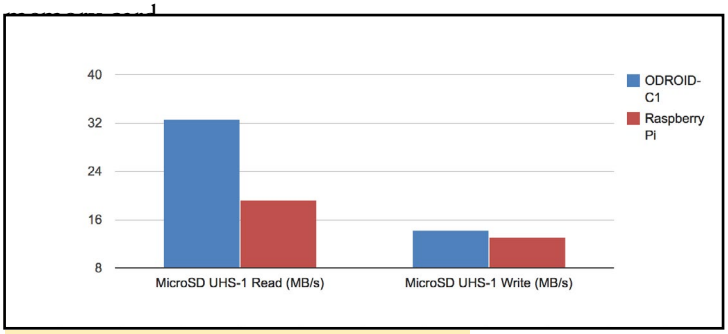
Storage I/O comparison

To obtain the results in the storage I/O comparison graph, type the following lines at a command prompt. The first command tests the write speed, and the second command tests the read speed:

```
$ dd if=/dev/zero of=test.tmp oflag=direct bs=500K
count=1024
$ dd if=test.tmp of=/dev/null iflag=direct bs=500K
count=1024
```

Media access performance	ODROID-C1	Raspberry Pi
eMMC Read (MB/s)	62.2	N/A
eMMC Write (MB/s)	25.1	N/A
MicroSD UHS-1 Read (MB/s)	32.5	19.2
MicroSD UHS-1 Write (MB/s)	14.2	13.1

If you use the eMMC storage, you can get about two to three times faster storage I/O read performance. However, an affordable microSD UHS-1 card can still achieve reasonably fast speeds thanks to the advanced SD 3.01 host controller in the S805 processor. The MicroSD card read performance on C1 is still about 1.7 times faster then RPi if you use the UHS-1 memory card.



File I/O comparison of CI vs Rpi

Ethernet I/O comparison

To replicate the results in the Ethernet comparison graph, type the following lines at a command prompt:

Server agent on the CI and Client agent on the host PC

```
ruppi@ruppi-desktop:~$ iperf -c 192.168.2.10 -P 10
[SUM]  0.0-10.0 sec   700 MBytes   584 Mbits/sec
```

Server agent on the host PC and Client agent on the CI

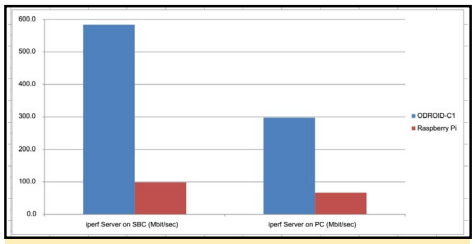
```
odroid@odroid:~# iperf -c 192.168.2.2 -P 10
[SUM]  0.0-10.1 sec   351 MBytes   292 Mbits/sec
```

Server agent on the RPi and Client agent on the host PC

```
ruppi@ruppi-desktop:~$ iperf -c 192.168.2.11 -P 10
[SUM]  0.0-10.3 sec   121 MBytes  98.6 Mbits/sec
```

Server agent on the host PC and Client agent on the RPi

```
pi@raspberrypi:~$ iperf -c 192.168.2.2 -P 10
[SUM]  0.0-10.3 sec  81.6 MBytes  66.6 Mbits/sec
```



Networking comparison of C1 vs Rpi

Thanks to the Gigabit Ethernet available on the ODROID-C1, the network performance of C1 is an impressive four to six times faster than Raspberry Pi.

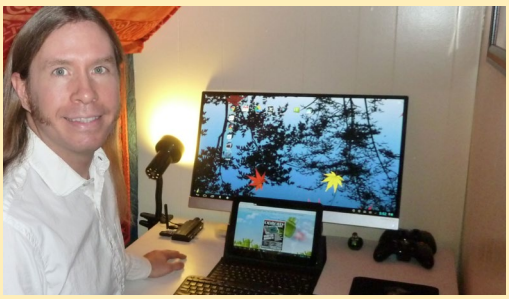
Conclusion

As one can see from the testing results, the quad-core 1.5GHz ODROID-C1 can easily outperform the single-core 700MHz Raspberry Pi board, even when the Raspberry Pi is overclocked. Many test results show four to seven times more performance from the ODROID-C1 platform.

Even though both platforms are Linux-friendly computing devices, the performance to cost ratio is much higher with the ODROID-C1. If you are considering a tiny computer for general purpose computing, software development, or as a project platform, the ODROID-C1 will give you a lot more satisfaction and fun with incredible performance for a very low price.

The Hardkernel team that makes it all happen

Rob Roy in his ODROID den with a Q2 and U3



BEHIND THE SCENES

THE AMAZING STAFF OF YOUR FAVORITE TECH MAGAZINE

by Rob Roy

Now that ODROID Magazine has completed its first full year of publication, we thought that it might be a good time to introduce you to the work process of the magazine staff that brings you the latest ODROID news, tips, and hacker projects every month.

We start by reading through the article submissions from our worldwide network of contributors, who send their submissions either through email or by posting on the ODROID forums. Rob Roy and Bo edit the articles for consistency and technical accuracy, then save the final versions to Google Drive.

Once the articles are completed, the art editors Bruno, Nicole and James carefully transcribe them using Adobe InDesign, and create a custom layout for each page by placing all of the article images and screenshots alongside the text. At the same time, Bruno creates the cover each month based on the feature article, which is usually written by the team from Hardkernel.

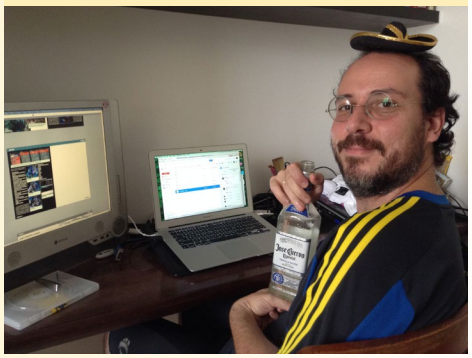
Next, we all come up with funny pictures and captions to fill up the extra space, with some of our favorite images of robots, cats, dogs, and other silly things. We hope you like our sense of humor.

Finally, Manuel takes the magazine contents and translates them into Spanish for our worldwide audience. Another job well done!

Bo also owns the Ameridroid distributor

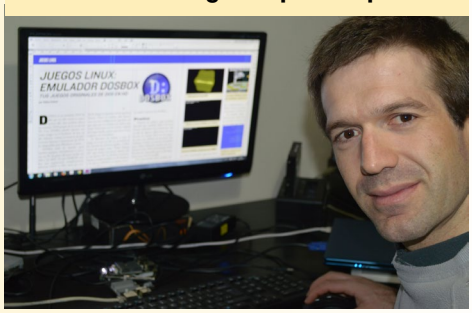
Nicole has a cozy office with lots of plants

Bruno with a fancy hat and bottle of tequila



James likes a natural setting for his work

Manuel is our bilingual expert in Spanish



GAMECADE

A MINIATURE ARCADE CONSOLE WITH AUTHENTIC CONTROLLERS

by @jrmago

The GameCade is... AWESOME!!!

In creating an ODROID miniature gaming system to run classic arcade games, my initial idea was to use Android and MAME4droid, but Android did not have good compatibility with the joysticks card that I bought. Instead, I ended up using AdvanceMAME running on Linux. To build the console, I used an interface kit to provide controllers and button for the system at http://bit.ly/11r0NhT. The rest is wood, screws, paint and patience!

In order to turn it into a coin-operated machine, I plan to use the premade unit at http://bit.ly/1wVO3bQ. I can either have a signup button for each person that allows the machine to know which player has paid, or just get four separate coin acceptors -- one for each player -- like most arcade machines have. Of course, for an office machine, you don't need to charge anybody, but the option is there for the authentic feel. Also, those coin acceptors are programmable, which means you can use any kind of coin or game token that will fit in the slot. To train it, put the coin acceptor in programming mode and run 20 of the types of coin that you want to use through it so it can learn the weight and size of the coin that it is supposed to accept. It pulses for a programmable length of time between 20-60ms for each coin it accepts, allowing it to be hooked up to any free GPIO.

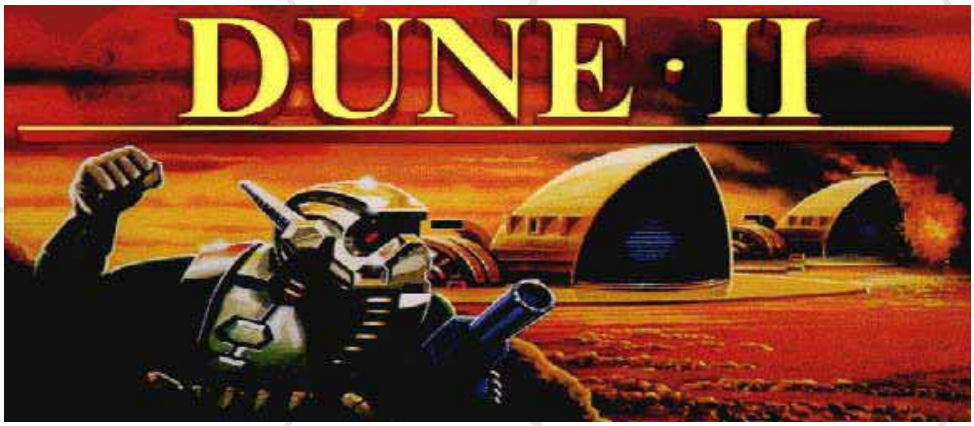
This arcade monster is ready to eat your quarters

The inner workings of the GameCade before assembly

LINUX GAMING

IMPROVED DOS REMAKES OF CLASSIC GAMES

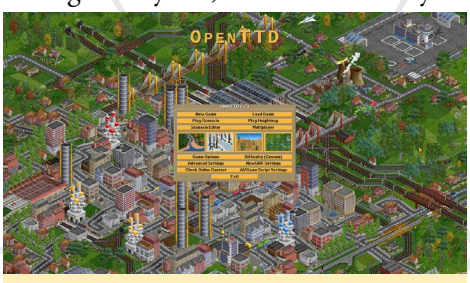
by Tobias Schaaf



Last month, I discussed DOSBox and emulation of old DOS games on the ODROID, and the conclusion was that although the ARM-optimized version of DOSBox runs well, allowing you to replay many of your favorite games, it still lacks good performance, and not every game runs smoothly on the ODROID system. This time, I want to focus on native Linux ports of different DOS games, and see what has changed between the original and the remake.

Transport Tycoon Deluxe

OpenTTD is a perfect example of a very successful game port which outshines the original without destroying the charm of the game. For those who do not know this game, it's part of the Tycoon series, which means that it's a economy simulation. In this particular one, you are the manager of a transportation company able to transport goods, resources and people via train, bus, truck, boat, or airplane. As you progress through the years, new modern way of transportation come up.



OpenTTD Title screen

Although you start with old steam powered locomotives, you later on are able to use electrical locomotives and monorails. The objective of the game is to build up a network of materials and goods that you transport. For example, you can transport coal from a coal mine to a power plant, or iron ore from an iron mine to a steel factory. In order to achieve this, you build tracks, roads and establish other ways of transportation. The game comes with different scenarios in different time spans, randomly generated worlds, different worlds (there is for example a candy world where you transport candy instead of coal, wood, or oil) and lots of goodies.

OpenTTD is a complete remake of the original DOS game with several added features. If you know and love the original game, you will find everything that you enjoy in this remake as well. You can even use your original game files to play the game with the older game graphics, but that's not required. Graphics, sounds and music were completely redone and are free and open source, and no original game files are required to play it.

The game was ported to many different systems. There are Windows, Linux, Mac, Android, iOS and Windows Phone ports of this game, and they all perform amazingly well. You can even play the game on the ODROID in 1080p without any issues.

OpenTTD differences

This game has so many new modifications that it's almost overwhelming. If you prefer to build rail tracks and rely mostly on trains, but don't want to play the game on easy mode, pick a computer opponent that concentrates on railroads as well so that you can experience a new challenge. For example, if you are from China, and don't want to see American or European town names, you can pick the Chinese Towns Name modification. If you are from Germany and want German signs and light signals, you can choose the German signal modification. If you are a fan of really old trains from the United States circa 1850, select the North American Renewal Set. Do you want helicopters in the game? Guess what, there's a modification for that too!

With new buildings, new vehicles, new street designs, additional content

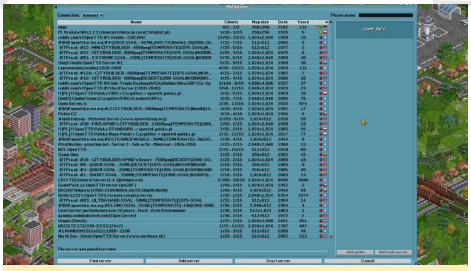
OpenTTD online modification list

and scenarios, you will find everything in the online content library that lets you create the game that you like. Other improvements include different resolutions, lots of different languages, varying currency and measurement systems, and you can even fine tune a lot of options like the behavior of cars, such as the rate at which they break down.

Another big advantage of OpenTTD is its cross-platform multiplayer mode, which means that you can play using the

Linux version on the ODROID while another family member plays it on their Windows PC, or even with an Android tablet. You not only can play the game at home in your local LAN, but also with others over the Internet.

You may ask yourself, how many people would actually play a old DOS game like this online? Take a look at the multiplayer screenshot! As you can see, there are still quite a few servers and games up



OpenTTD multiplayer Internet game servers

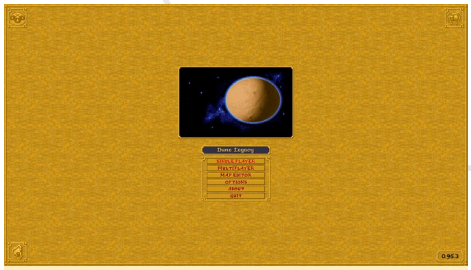
and running, and the game is very fun to play and explore. It's always exciting to earn the first million (and later billion) dollars on your account. You aren't even required to fight each other in multiplayer. Different people can control the same company in the game, and you can literally be in different places at the same time.

OpenTTD supports up to an astounding 255 simultaneous players, which can be organized into as many as 15 different companies. Imagine a game of 200 people with just two companies consisting of 100 players, cooperating against 100 other cooperative players, which is just amazing! Maps can be a very large 2048x2048 units (and can be even bigger with some modifications), which is actually big enough to support 255 players.

If you like simulation games like this, OpenTTD is definitely a must have!

Dune 2 – Dune Legacy

In the previous DOSBox article, I mentioned Dune 2, which is the grandfather of all Real Time Strategy games, published by the same company that later created the awesome Command and



Dune Legacy title screen

Conquer series. Dune Legacy is a re-implementation of the Engine for Dune 2, which means that it still requires the original game data to play the game.

Dune Legacy offers some nice improvements over the standard version of the game. Some of the improvements can be found in the options menu, such as different resolutions and some perfor-

Dune Legacy option menu

mance scalers. Others can be found in a sub-menu that allows you to alter the default game mechanics to create a different kind of gaming experience. Besides that, the game itself was changed in several ways that really enhanced the gameplay.

One of the biggest improvements over the original game is the option to draw a select box around multiple units and command all of those units together. In the standard version, you could only select and command a single unit. Sending out an army was rather difficult and hard on the nerves, since the first units often reached the target before you were done sending the last units on the way, depending on the size of the army that you wanted to deploy.

Another improvement is that you now can direct units by simply right clicking at a spot, and attack a unit by right-clicking the enemy unit. This may sound strange to some people, but the original game did not offer this simple feature. You actually had to either click the "move" command, or press the "M" key on the keyboard to send a unit away. This made the game very difficult when there was a lot of action going on on the screen. However, these features are now available on Dune Legacy.

Other improvements can be seen in the building menu, which was completely redesigned and now offers access to all of the available items right on the screen. In the original version, you had to switch between build options by clicking on the building itself, clicking on the icon of the building to open the menu in order to select what you want to build, selecting the desired unit or building, then clicking the OK/select button.

If you wanted to construct something else after that, you had to wait until the first build was done, and then select the next item by going through all of the steps again. You could also only build

Dune Legacy new building menu

one unit, meaning you had to wait until the unit was done building before you could start building the next unit.

With Dune Legacy, this has all changed. You now have a Command and Conquer-style building menu, allowing you to see all the build options by clicking a building. There is also a build queue where you can select different units or buildings after another, and start building the next one right after the previous is completed. You can even select how many units that you want to build. Creating a queue with 10 tanks is rather easy now.

Dune Legacy offers some other new features, like improved behavior of the units, where you can tell them to either guard the spot where they are standing, or chase after an enemy. It also has a multiplayer LAN and Internet mode. On skirmish and multiplayer maps, you can play as Fremen, Sardurka and Mercenaries as well as the original houses, giving the game a special twist.

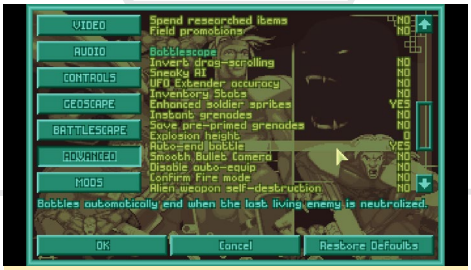
All in all, I really like the improvements of Dune Legacy, pushing Dune 2 down a notch on the list of my favorite RTS games.

UFO: Enemy Unknown – OpenXCom

Another item that I mentioned in my last article was the X-Com series, in which you are humanity's last stand against an evil alien species trying to conquer earth. This game is huge, with numerous options and tasks within the game.

OpenXCom is essentially a turn-based strategy (TBS) game, although it's actually more than that. There are economical aspects to the game, where you have to organize your funds and distribute your money and resources over different projects by evaluating what's most important to you. You can research new technologies or make an autopsy on an alien, manufacture weapons, ammunition, med packs and other items. You also have to manage your soldiers, train them, equip them with weapons, armors, movement detectors and more. Other parts of the game include hunting and shooting down UFOs, and sending your soldiers on missions to either help the human population or recover artifacts from a crash site of an UFO. This is when the game switches over to a turn-based mode, and you have to send your men one by one into combat.

OpenXcom title screen



OpenXcom advanced options

The game has received some improvements as well, which you already can see when you open up the option screen, as shown in the screenshot. There are a lot of options here which were never in the original game. You can change the resolution, set display filters, and choose if you want to play in window or fullscreen mode. The scaling of the Geoscape and Battlescape defines the size of the menus, and how much the battle field is scaled. The higher the value, the smaller the menu, which allows you to see more of what's happening on the screen, but it also makes everything smaller, up to a point where you can't see anymore what's actually going on.

OpenXcom options

Some of the features are unfortunately unsuitable for the ODROID, since they were written for OpenGL, which is not available on the ODROID platform (which uses OpenGLES), but the game works fine without these features. Under the "Advanced" menu, you can change several aspects of the game. For example, you can automatically end a mission once all of the enemies are beaten, whereas in the original DOS version, you found out if you finished the mission only when you ended your turn. This could be annoying if you killed the last enemy at the beginning of your turn and then planned the movements of another 10 squad members trying to find more aliens, just to find out that everything that you did in the last 10 minutes was pretty much useless. So this feature is a very welcome improvement.

There are other useful options such as the possibility to end a mission if you "capture" an enemy with psionic mind control, rather than needing to wait until the mind control kills the enemy. The options should be checked carefully since they can really improve your gaming experience.

The game itself is pretty close to the original, and includes some great improvements, especially in the management of the base. A very nice feature is the ability to directly sell the items you can produce at your base, which steadily increases your income. A lot of tiny

OpenXcom geoscape

changes were made which come very handy, such as the ability to right-click on the arrow in the research menu, allowing you to add all of the available scientists at once rather than individually, and an option to manufacture unlimited items of a certain kind.

The game has so many aspects and incremental improvements that I can't make pictures of all of them or even mention them all. For example, there are even modifications that let you play with pirate characters and other crazy things.

I really like OpenXCom, and it ranks as one of my all-time favorites. OpenXcom is a remake of the first X-Com game (UFO: Enemy Unknown), but they are working on the second game as well (Terror From The Deep). Over the years, the XCom series has had a few sequels, such as the "UFO" series (UFO – Aftermath, UFO – Aftershock, UFO – Afterlight) that aren't well known. They are not turn-based, but happen in real time with an intelligent pause mode, and I love playing them because they still offer similar gameplay to the original, such as researching, manufacturing, and caring for your soldiers. I highly recommend the entire X-Com series of games.

UFO: Enemy Unknown received an official remake about a year ago (in 3D!) but was still a turn-based game. It has already been ported to Android, which means that it's successfully running with OpenGLES 2.0. I look forward to seeing this game available on the ODROID Linux platform, but I doubt that it will happen anytime soon.

OpenTTD, Dune Legacy and OpenXCom are just some of my favorite games that have been ported from DOS to Linux. Next month, I will present some more games that capture the excitement of the original versions while offering modern improvements.

FUN WITH RTL-SDR

FM RADIO AND REAL-TIME FLIGHT TRACKING

by Venkat Bommakanti

Have you ever wondered how you can pass extra time during your next long layover at the airport? We have a fun project that will come to your rescue! See what you can do with a mini portable laptop (U3, VU/Lapdock, 2A battery-packs, and wireless keyboard/mouse) and an inexpensive (US$10) RTL-SDR dongle, such as http://amzn.to/1udetDW and http://bit.ly/1dtzV0T.

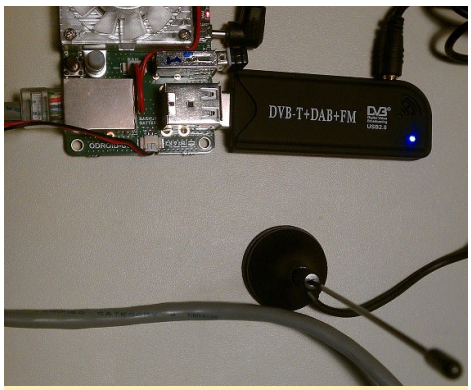
Requirements

1. An ODROID-U3 board, with an appropriate power adapter.
2. A Class 10 MicroSD (with an SD card reader/writer) with the latest U3-specific Lubuntu desktop desktop image, such as ubuntu-14.04.1lts-lubuntu-odroid-u-20140814.img and an 8+ GB eMMC card.
3. A network where the device has access to the Internet and the ODROID forums.
4. SSH access to the U3 via utilities like PuTTY (MS Windows 7+), terminal (Mac, linux), etc., from the remote desktop.
5. An RTL2832U/R820T based USB2 RTL-SDR dongle such as the one listed in the reference (links) section.
6. Latest open-source Software Defined Radio (SDR) rtl-sdr library from OsmoSDR.
7. A working version of the dump1090 software.

Preparation

Boot up the U3 with the latest Lubuntu desktop software. Update the system by selecting all the menu options of the ODROID utility, then shutdown the system and power it off. Attach the USB2 RTL-SDR dongle to the U3 as shown in the screenshot, then attach headphones as well.

After connecting the hardware, power on the system. Once the desktop appears, check to see whether the RTL-SDR dongle has been detected by Linux using the lsusb command in a Terminal window. The output should appear as follows:



RTL-SDR setup

```
odroid@u3-1:~$ lsusb
Bus 001 Device 005: ID 0bda:2838
Realtek Semiconductor Corp.
RTL2838 DVB-T
```

Note the ID of 0bda:2838 (VID:PID) of this device - it happens to be that of a device supported by the RTL-SDR library of OsmoCom (http://bit.ly/1fRNBHs). You will have to obtain a supported RTL-SDR device for this project to work. What is common in many device categories is that different manufacturers' devices may use the

same VID:PID combination, and one may work but another may not. Also note the bus:device-id of 001:005 in the output. Using that information, the following command outputs detailed device information for verification:

```
odroid@u3-1:~$ lsusb -D /dev/bus/
usb/001/005

Device: ID 0bda:2838 Realtek
Semiconductor Corp.  RTL2838 DVB-
T
Couldn't open device, some infor-
mation will be missing
Device Descriptor:
  bLength             18
  bDescriptorType              1
  bcdUSB              2.00
  bDeviceClass                 0
(Defined at Interface level)
  bDeviceSubClass              0
  bDeviceProtocol              0
  bMaxPacketSize0     64
  idVendor            0x0bda Re-
altek Semiconductor Corp.
  idProduct           0x2838
RTL2838 DVB-T
  bcdDevice           1.00
  iManufacturer                1
  iProduct                     2
  iSerial                      3
  bNumConfigurations  1
  Configuration Descriptor:
      bLength                  9
      bDescriptorType          2
      wTotalLength             34
      bNumInterfaces           2
      bConfigurationValue      1
      iConfiguration           4
      bmAttributes             0x80
      (Bus Powered)
      MaxPower
500mA
      Interface Descriptor:
      bLength                  9
      bDescriptorType          4
      bInterfaceNumber         0
      bAlternateSetting        0
      bNumEndpoints            1
      bInterfaceClass          255
Vendor Specific Class
      bInterfaceSubClass       255
Vendor Specific Subclass
      bInterfaceProtocol       255
Vendor Specific Protocol
      iInterface               5
      Endpoint Descriptor:
      bLength                  7
      bDescriptorType          5
      bEndpointAddress         0x81
EP 1 IN
      bmAttributes             2
              Transfer Type
Bulk
              Synch Type
None
              Usage Type
Data
      wMaxPacketSize  0x0200  1x
512 bytes
      bInterval                0
      Interface Descriptor:
      bLength                  9
      bDescriptorType          4
      bInterfaceNumber         1
      bAlternateSetting        0
      bNumEndpoints            0
      bInterfaceClass          255
Vendor Specific Class
      bInterfaceSubClass       255
Vendor Specific Subclass
      bInterfaceProtocol       255
Vendor Specific Protocol
      iInterface               5
```

However, examining the dmesg logs, you'll see that the appropriate driver is not loaded.

```
odroid@u3-1:~$ dmesg | grep dvb
[  232.967274] usb 1-3.2: dvb_
usb_v2: found a 'Realtek RTL2832U
reference design' in warm state
[  232.967407] usbcore: regis-
tered new interface driver dvb_
usb_rtl28xxu
[  233.029934] usb 1-3.2: dvb_
usb_v2: will pass the complete
MPEG2 transport stream to the
software demuxer
[  233.030736] usb 1-3.2: dvb_
usb_rtl28xxu: unknown tuner=NONE
[  233.042345] usb 1-3.2: dvb_
usb_v2: 'Realtek RTL2832U refer-
ence design' error while loading
driver (-19)
[  233.042905] usb 1-3.2: dvb_
usb_v2: 'Realtek RTL2832U refer-
ence design' successfully deini-
tialized and disconnected
```

On further examination, you can see that a relevant kernel module, which is part of the installed image, is dvb_usb_rtl28xxu. In this example, only one of the RTL modules worked:

```
odroid@u3-1:/lib$ sudo find .
-name *rtl*
...
./modules/3.8.13.28/kernel/driv-
ers/media/usb/dvb-usb-v2/dvb-usb-
rtl28xxu.ko
./modules/3.8.13.28/kernel/driv-
ers/media/dvb-frontends/rtl2832.
ko
./modules/3.8.13.28/kernel/driv-
ers/media/dvb-frontends/rtl2830.
ko
...

odroid@u3-1:~$ sudo modprobe dvb_
usb_rtl2832
modprobe: FATAL: Module dvb_usb_
rtl2832 not found.

odroid@u3-1:~$ sudo modprobe dvb_
usb_rtl2830
modprobe: FATAL: Module dvb_usb_
rtl2830 not found.

odroid@u3-1:~$ sudo modprobe dvb_
usb_rtl28xxu
```

Prerequisites

While it is presumed that the system is up to date, it is worthwhile to run the following commands to install any missing components. If using the Lubuntu image supplied by Hardkernel, it will probably already include these applications:

```
$ sudo apt-get install git-core
git cmake
$ sudo apt-get install libusb-
1.0-0-dev build-essential
```

Build the library software

Change to your home directory and fetch the source code from git repository using these commands:

```
$ cd ~
$ git clone git://git.osmocom.
org/rtl-sdr.git
```

The source is placed in a newly created rtl-sdr sub-directory. Now, prepare to build the source code using the following commands:

```
odroid@u3-1:~$ cd rtl-sdr/
odroid@u3-1:~/rtl-sdr$ mkdir
build && cd build
odroid@u3-1:~/rtl-sdr/build$
cmake ../ -DINSTALL_UDEV_RULES=ON
odroid@u3-1:~/rtl-sdr/build$ make
odroid@u3-1:~/rtl-sdr/build$ sudo
make install
...
[  5%] Built target convenience_
static
[ 35%] Built target rtlsdr_shared
[ 40%] Built target rtl_adsb
[ 45%] Built target rtl_eeprom
[ 50%] Built target rtl_fm
[ 55%] Built target rtl_power
[ 60%] Built target rtl_sdr
[ 65%] Built target rtl_tcp
[ 70%] Built target rtl_test
[100%] Built target rtlsdr_static
```

Configure the dynamic linker runtime bindings using the command:

```
odroid@u3-1:~/rtl-sdr/build$ sudo
ldconfig
```

Check for the presence of the rtl-sdr utilities just built:

```
odroid@u3-1:~/rtl-sdr/build$
which rtl_eeprom
/usr/local/bin/rtl_eeprom
```

Examine the dongle's eeprom contents. Note that this device variant uses the Rafael Micro R820T tuner.

```
odroid@u3-1:~/rtl-sdr/build$ rtl_
eeprom
Found 1 device(s):
  0:  Generic RTL2832U OEM

Using device 0: Generic RTL2832U
OEM
Found Rafael Micro R820T tuner

Current configuration:

_____________________________________
__________
Vendor ID:
0x0bda
Product ID:
0x2838
Manufacturer:                Re-
altek
Product:
RTL2838UHIDIR
Serial number:
00000001
Serial number enabled:       yes
IR endpoint enabled:         yes
Remote wakeup enabled:       no
_____________________________________
__________

odroid@u3-1:~/rtl_sdr$ rtl_test
-t
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN:
00000001

Using device 0: Generic RTL2832U
OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0
0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4
15.7 16.6 19.7 20.7 22.9 25.4
28.0 29.7 32.8 33.8 36.4 37.2
38.6 40.2 42.1 43.4 43.9 44.5
48.0 49.6
Sampling at 2048000 S/s.
No E4000 tuner found, aborting.
```

Requirements

In the list of utilities just built, of special interest is rtl_fm, which is basically an FM radio tuner. Prior to any listening test, such as FM reception, it is always safe to ensure that the audio volume levels are safe, otherwise you could damage your hearing. First, disconnect all headphones or speakers, adjust the volume to 5% of maximum, then reconnect them.

Next, select an FM station closest to you. In my case, it would be the local Classic Rock favorite KFOG 97.7FM. Using the station frequency, launch the FM tuner like so:

```
odroid@u3-1:~/rtl-sdr$ rtl_fm -f
97.7M -M wbfm -s 200000 -r 48000
- | aplay -r 48k -f S16_LE
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN:
00000001

Using device 0: Generic RTL2832U
OEM
Found Rafael Micro R820T tuner
Tuner gain set to automatic.
Tuned to 98016000 Hz.
Oversampling input by: 6x.
Oversampling output by: 1x.
Buffer size: 6.83ms
Sampling at 1200000 S/s.
Output at 200000 Hz.
Playing raw data 'stdin' : Signed
16 bit Little Endian, Rate 48000
Hz, Mono
underrun!!! (at least 326.562 ms
long)
```

Note the syntax for the frequency specification (97.7M) and internal conversion (to 98016000), using MHz and Hz units respectively. If you are using headphones, get them close to your ear.

You should be able to hear the selected radio station. After ensuring the volume level is safe, wear them and increase the volume level to a safe desired value.

You may observe some static, but the dongle is only $10, which is great for the price. More expensive dongles may yield better audio quality. You can scan many more varieties of FM transmissions such as police and pagers, but please comply with your local laws to ensure you are not violating them. More information about setting up the FM receiver may be found at http://bit.ly/1uU8mK1.

Real-time flight view

If you thought that listening to an FM radio station via your U3 was cool, there is an even cooler activity - tracking flights in your vicinity! Essentially, you can track them using a feature of this class of dongles - Automatic Dependent Surveillance-Broadcast (ADS-B) - by tuning the device to the standardized 1090 MHz frequency using the in-built Mode S decoder. All one has to do is interpret the data that is transmitted using the dump1090 application.

Please note that the information presented here is meant for entertainment use only. Again, please comply with your local laws to ensure you are not violating them while tracking planes or using radio frequencies. Hardkernel and the magazine content providers are not responsible for violations.

Build dump1090

Fetch and build the dump1090 application using the commands:

```
odroid@u3-1:~$ git clone git://
github.com/antirez/dump1090
odroid@u3-1:~$ cd dump1090/

odroid@u3-1:~/dump1090$ export
PKG_CONFIG_PATH=/usr/local/lib/
pkgconfig/

odroid@u3-1:~/dump1090$ make

odroid@u3-1:~/dump1090$ export
LD_LIBRARY_PATH="/usr/local/
lib/:${LD_LIBRARY_PATH}"
odroid@u3-1:~/dump1090$ ./
dump1090 --interactive --net
```

This application comes with a web page (gmap.html), that maps the flights in real-time onto a google map of the local area selected. Since I live in the San Francisco area, I've chosen a latitude and longitude of 37 deg N and -122 deg W. Accordingly, the modified section of the web-page looks like this:

```
CenterLat=37.0;
CenterLon=-122.0;
```

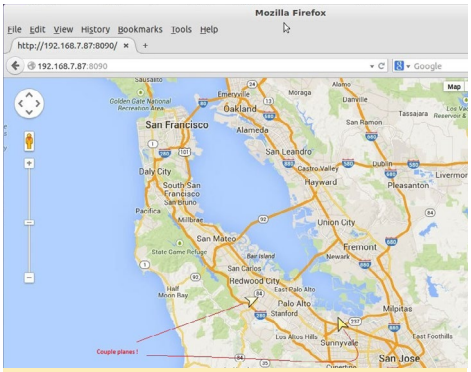
Track 'em

Place the antenna of the dongle away from interferences. You can now track flights using the web-page served by a built-in (dump1090) web server. To do so, stop listening to the FM receiver (if it's running), then launch the application using the command:

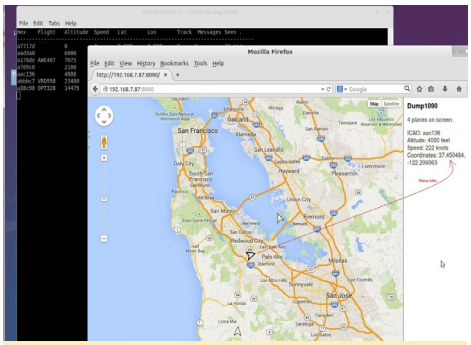
```
$ ./dump1090 --interactive --net
--net-http-port 8090

Found 1 device(s):
0: Realtek, RTL2838UHIDIR, SN:
00000001 (currently selected)
Found Rafael Micro R820T tuner
Max available gain is: 49.60
Setting gain to: 49.60
Exact sample rate is:
2000000.052982 Hz
Gain reported by device: 49.60
…
```

The built-in web server uses port 8090. Launch a browser on the U3 and point it to http://127.0.0.1:8090. In a few moments you should see a mapped view of flights nearby (presuming you are living near airplane flight paths), as shown in the screenshot. For more information about dump1090, please refer to http://bit.ly/1xF0VbV, http://bit.ly/1pqCuWi and http://bit.ly/1yrwCSZ.

Click the link in top right corner of the web page to see flight details, which can then be compared with the centralized map at http://planefinder.net.



Map of several planes being tracked by an ODROID-U3 with an RTL-SDR dongle



Viewing a flight using ADS-B (Automatic Dependent Surveillance Broadcast)

ZX SPECTRUM

A HANDHELD RETRO GAMING DEVICE

by Daniel Bull

I'd been wanting to build a handheld ZX Spectrum for a while, and was thinking of using a Raspberry Pi running FUSE (Free Unix Spectrum Emulator) but the Pi was just too big, so when the ODROID-W was announced, complete with LiPo circuits, I decided that now was the time! I started making layouts using color printouts, since the ODROID-W wasn't yet available for sale.

I started with a PiTFT screen (http://bit.ly/1zDmHL5), which was actually running from a Raspberry Pi at this point, with some custom PCB's either side and below. This was followed by two DECT portable phone batteries, along with some old Dell laptop speakers and the printout of the ODROID. I used a USB sound card dongle which was originally intended to power the speakers, although I eventually ditched this, since I ended up using the USB for WiFi instead.

One of the things which took the longest was working out a set of keys to make up the controls that were compatible with the maximum number of games. This is because early Spectrum games rarely used the same keys, and joysticks were uncommon and came in a few different styles.

Next, I created some designs using 123D Design for my 3D printer. It's built using a front and back piece, with both being as compact as possible in order to keep it small and light. Those of you that have used a Spectrum will recognize the space for the stripes on the front.

Then, I printed the back piece and started to layout the components, and thankfully they all fit!

It did actually take me two attempts, since I had problems with warping on the first try, but this gave me a chance to tweak the layout to perfection. The small PCB above the right speaker is actually for a power/reset button which is rear facing.

I started wiring, and quickly realized that due to the large number of buttons, which used all of the GPIOs, the front and back weren't going to come apart very easily. Since my original plan for upgrading the software was to remove the SD card, it was obvious that this wasn't going to be practical. That's when I decided to get rid of the USB soundcard in order to free up the USB port, and instead used a small amp for the speakers. The original reason for the soundcard was that the output from the Broadcom wasn't enough to drive the speakers directly. Freeing up the USB port meant that I could install a WiFi dongle and access the project remotely for upgrades.

I used a standard Edimax WiFi dongle and dismantled it in preparation for mounting on the ODROID. I wrapped it in Kapton tape, which is something that those with 3D printers will likely have lying around, since it's used on the print beds, and attached it to the ODROID. I hooked it up to the USB, then assembled the entire unit. It works perfectly!

Materials

The shoulder switches are from eBay (http://ebay.eu/1uJJDrS), and it looks like the part number might be V5 SW052 MS-118, but there is nothing written on them. I got the heavy duty version to make sure they spring back properly.

I'm getting really good life out of the batteries, which are two packs of 2x950mAh that total 3800mAh. I've actually measured them at 3400mAh. So, if we say 3400mAh at ~4V, that comes to 13,600mW, which works out to 2720mAh at 5V. I'm not sure what power the ODROID will draw while running the emulator, but the WiFi is 40mA, and the screen is 100mA, so if the ODROID uses ~150mA, that means around 9 hours of battery life, which is very good.

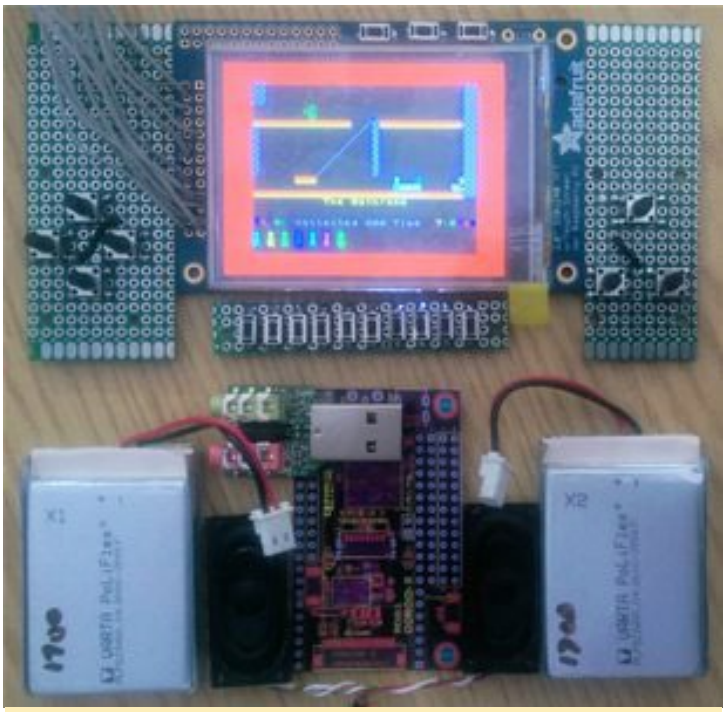
I got a 3D printer kit because, although I have quite good mechanical and electronic skills, I didn't know much about 3D printing. I chose a Mendel 90 by NopHead (http://bit.ly/1uwHhLg), which costs UK£500+VAT (UK). I don't regret spending that amount at all, since it was really well designed and worked on the first try. Also, the building process taught me a lot about how 3D printers work, and now I have learned enough to make my own changes and improvements.

I had never used CAD before either,

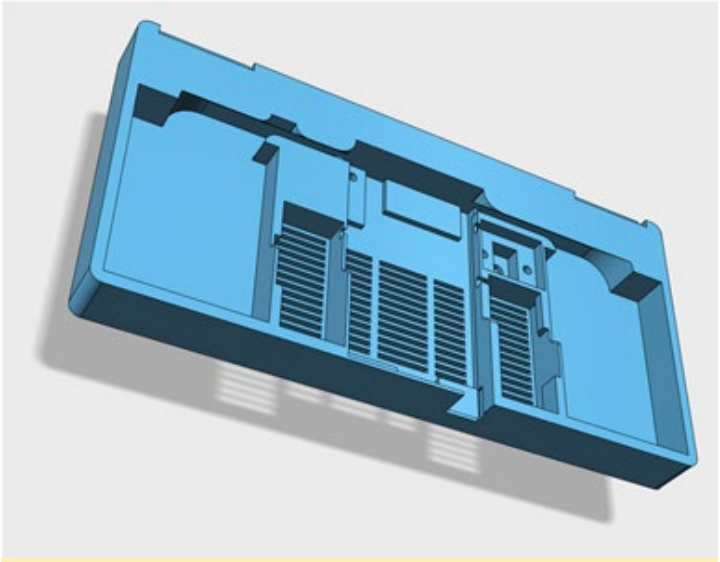
and I have to say it did take me quite a bit of free time to become familiar with it. I evaluated 3-4 different applications before settling on 123D Design. It was all worth it in the end though as you can see. If you don't own a printer, lots of places will print it for you, such as Shapeways (http://www.shapeways.com)

You can see more pictures of the printer, the build process, some prints, and my various modifications on my Google+ account at http://bit.ly/14MoNyj. I also recommend this video by Ben Heck, who also worked on his own Spectrum project, in order to help learn more about the CAD application: http://bit.ly/1zDpljH.

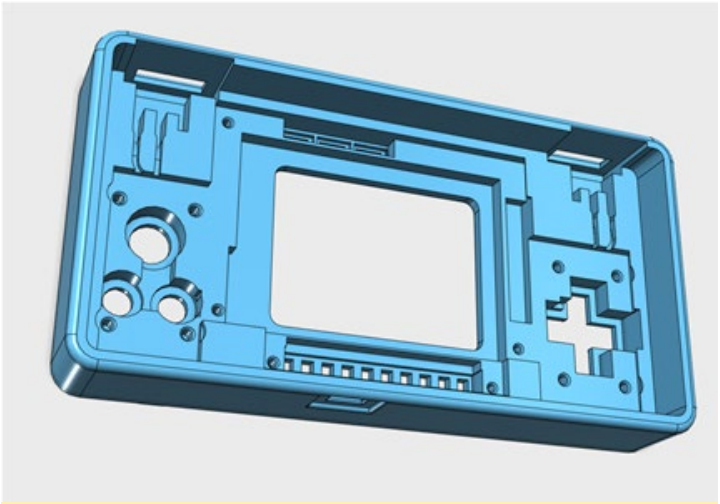
Top view of the Spectrum gaming machine



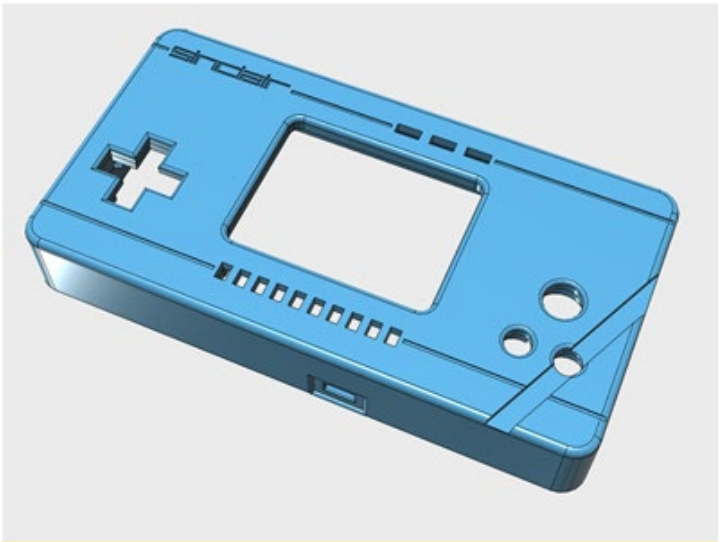
The Spectrum internal board, display, and batteries



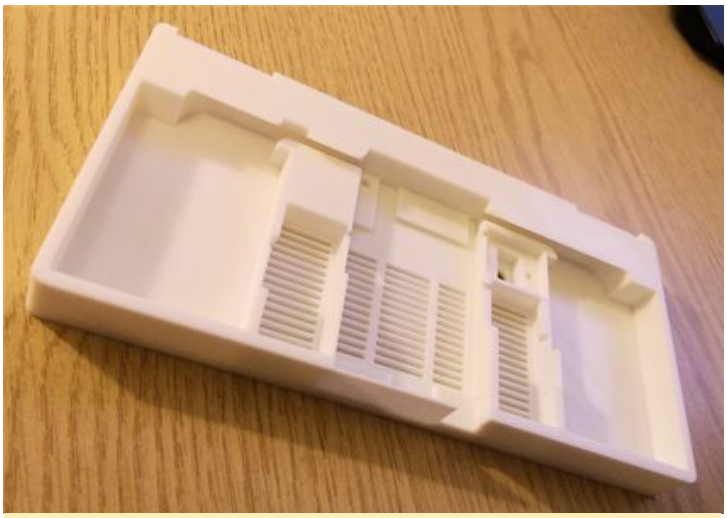
The 3D printed prototype of the case bottom



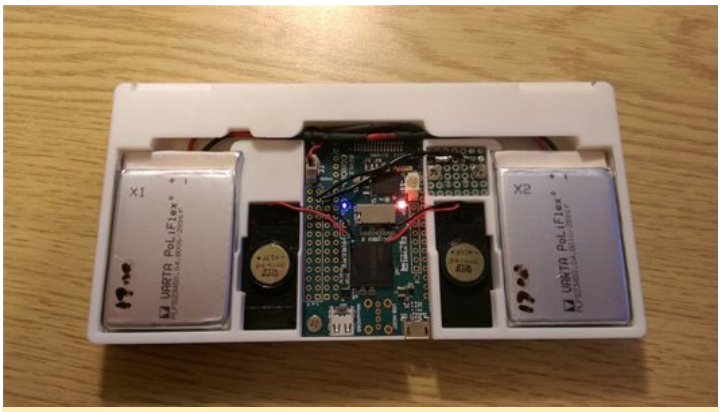
The design view prototype of the case top



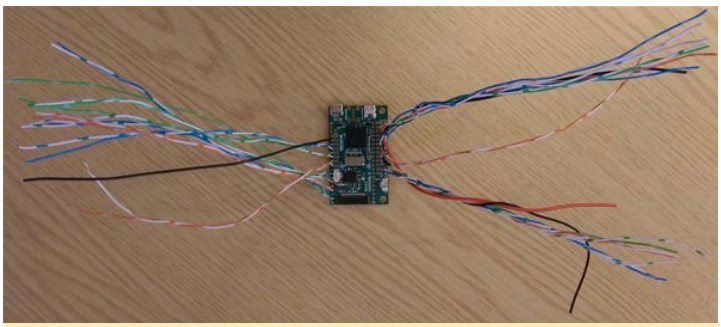
Top view of the 3D printed prototype of the case top



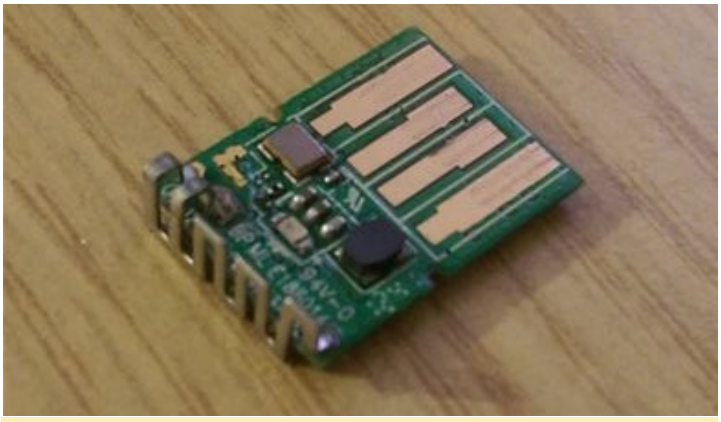
The case after being sent to the printer



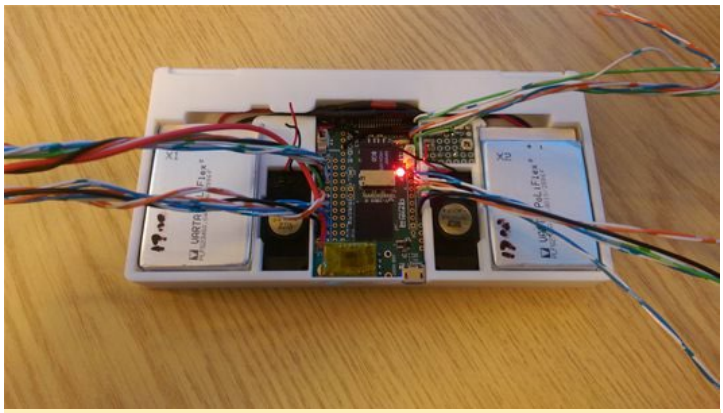
Partially assembled case without top



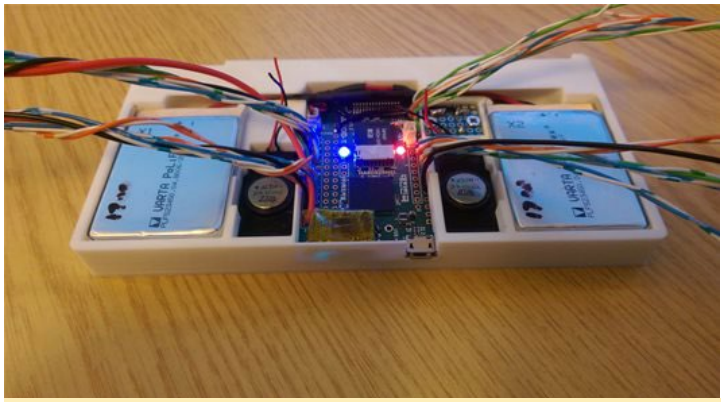
The I/O controller board with wiring



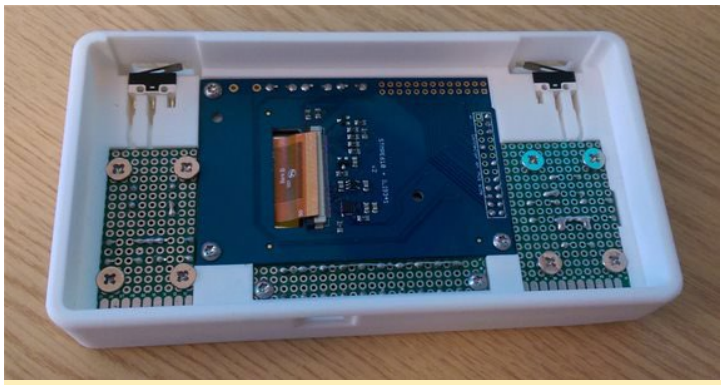
View of the WiFi dongle before assembly



The I/O controller placed in the case



Another view of the I/O controller



Inside of the top case before assembly

The completed assembly!

ANDROID DEVELOPMENT

ADDING BOOT ANIMATION TO THE INITIALIZATION PROCESS

by Nanik Tolaram

Android, like any other Linux based system, performs initialization using an init application. The init application is the first entry point to the user space after the kernel boots up, and is where Android launches its startup processes. The init application resides inside the system/core/init folder.

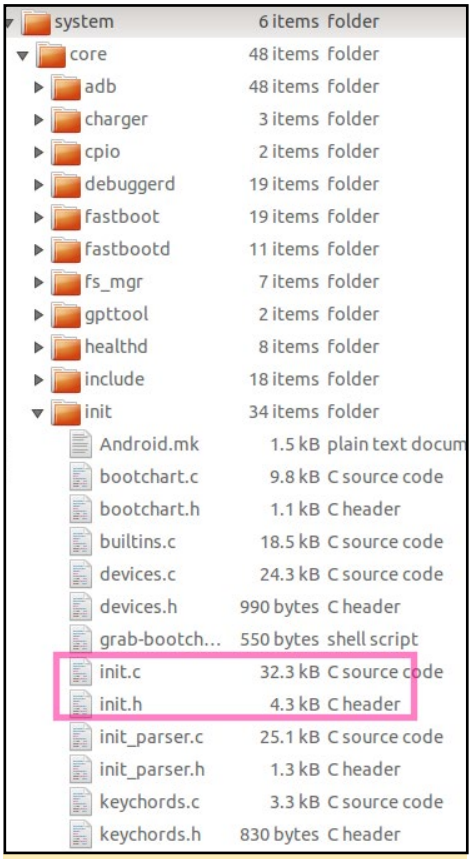


Figure I - Source location for init

The init application parses files with the .rc extension, and there can be several of them depending on the specific device. Like all configuration files, the .rc file is a plain text file with specific commands. We will drill down further in the next sections and look at these files.

.RC files

Most of the time, the .rc files reside inside the device/ folder. As can be seen in Figure 2, on an ODROID-U3, they are found inside the devices/hardkernel/odroidu/conf folder, including the init.odroidu.rc and init.odroidu.usb.rc files. The core init.rc file resides inside th /system/core/rootdir directory. The way in which these .rc files are linked together is as follows:

```
init.rc → init.odroid.rc → init.
odroid.usb.rc
```

Inside init.rc, there is an import statement at the top of the file:

```
import /init.usb.rc
import /init.${ro.hardware}.rc
```

The ${ro.hardware} section is a variable that is substituted during the build process, which, in the case of an ODROID-U3, is set to the value "odroidu", which means that the final file contains the following import statement:

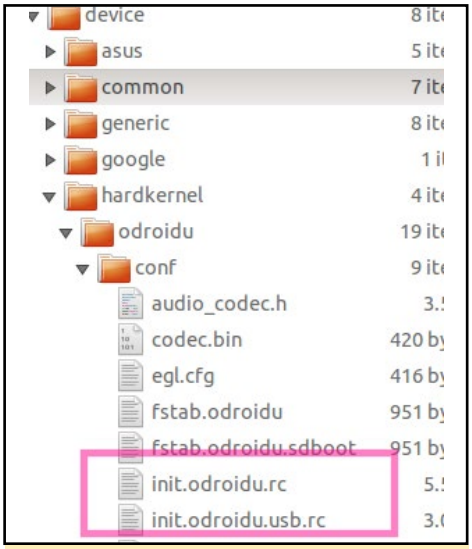


Figure 2 - Location of .rc files

```
import /init.odroidu.rc
```

At the top of the init.odroid.rc file is the line:

```
import init.odroidu.usb.rc
```

You can add your own .rc files and use the import statement to package them together. The .rc files follow a simple notation that is easy to learn, which can be broken down into the following categories:

Actions

These are actions that need to be performed during the init process. If you look inside init.rc, will see the following:

```
on early-init
    write /proc/1/oom_adj -16
    setcon u:r:init:s0
    start ueventd
    mkdir /mnt 0775 root system
```

The "on early-init" statement instructs the init application to execute the commands underneath it in the first stage of the initialization process. The commands will be executed when the init application starts up, and will do the following:

- write the number -16 to /proc/1/oom_adh
- execute setcon app using with the u:r:init:s0 parameter
- start up the ueventd daemon
- make directory /mnt with the parameter 0775 root system

Services

Services are applications that will be run during the init process. The init process along with other internal modules (including property service) will take care of stopping or restarting applications on behalf of the system or user. If you look inside init.odroidu.rc, you will see the following service definition:

```
service dhcpcd_eth0 /system/bin/
dhcpcd -ABDKL
    class main
    disabled
    oneshot

service iprenew_eth0 /system/bin/
dhcpcd -n
    class main
    disabled
    oneshot
```

The above statement is defining a service called dhcp_eth0, which is linked to the dhcpcd application. The iprenew_eth0 label is linked to the same dhpcd application, but with different parameters. The defined services will be run as background process. If you run ps or pstree, you will see the different applications defined as services running in memory.

The parameters below the service – class main, disabled and oneshot – are the properties of the service itself. Oneshot property tells init not to restart the application if it exits, disabled means the service is disabled or not running, and class main means the service belongs to a group called main. Normally services that are disabled will be run when certain conditions are met, such as property value changes. Grouping services using classes is useful if we need to start or stop applications as a unit.

Commands

There are specific commands that you can run when defining actions:

```
on boot
    mount debugfs /sys/kernel/de-
bug /sys/kernel/debug
    setprop ro.radio.noril yes
    write /proc/sys/vm/lowmem_re-
serve_ratio "128 128"
    chmod 0222 /sys/kernel/debug/
tracing/trace_marker
    write /sys/kernel/debug/trac-
ing/tracing_on 0

on fs
    mount_all /fstab.odroidu
    setprop ro.crypto.fuse_sdcard
true
```

In the above example, extracted from init.odroidu.rc, we can see that under the "on boot" action, we instruct the init app to execute the following commands: mount, setprop, write and chmod. The commands are the same as normal Linux utilities, but sometimes it varies in terms of what parameter need to be passed to it.

Properties

One of the ways in which the different modules in Android share information is by using properties. If you execute the command getprop inside Android (using Terminal Emulator or by typing "adb shell"), you will see the different properties that govern Android's internal settings.

Property Service

Most operating systems make use of environment variables to communicate states to different processes, and Android is no different. If you use the Android Debug Bridge (ADB) shell to monitor the ODROID-U3, you will see something similar to the output below when running the command "getprop" on Android's command line:

```
[dalvik.vm.dexopt-flags]: [m=y]
[dalvik.vm.heapgrowthlimit]:
[192m]
…
…
[dhcp.eth0.pid]: [1799]
[dhcp.eth0.reason]: [PREINIT]
[dhcp.eth0.result]: []
…
…
[init.svc.dhcpcd_eth0]: [running]
[init.svc.drm]: [running]
[init.svc.fuse_sdcard0]: [run-
ning]
[init.svc.healthd]: [running]
[init.svc.insmod_ax88179]:
[stopped]
[init.svc.insmod_smsc95xx]:
[stopped]
[init.svc.insmod_usb_audio]:
[stopped]
[init.svc.insmod_usbmidi]:
```

```
[stopped]
…
…
[ro.board.platform]: [exynos4]
[ro.build.characteristics]: [tab-
let]
[ro.build.date.utc]: [1414813591]
[ro.build.date]: [Sat Nov  1
14:46:31 EST 2014]
[ro.build.description]: [odroi-
du-eng 4.4.4 KTU84Q eng.nan-
ik.20141101.144528 test-keys]
[ro.build.display.id]: [odroi-
du-eng 4.4.4 KTU84Q eng.nan-
ik.20141101.144528 test-keys]
…
…
[ro.build.version.incremental]:
[eng.nanik.20141101.144528]
[ro.build.version.release]:
[4.4.4]
[ro.build.version.sdk]: [19]
[ro.product.brand]: [Android]
```

The way to modify environment variables is different in Android when compared to Linux. In Android, you use the commands setprop and getprop, which sets and gets properties. Try running "getprop ro.board.platform", and you will see the output of your board platform. The prefix that is used in the environment variables is significant. For example, the prefix ro means read only, so that the environment variable cannot be modified

The process that takes care of setting and getting the environment variables is called property_service, and it is run as part of the init process. If you browse through the /dev/socket directory, you will see the property_service socket:

```
srw-rw---- system   system
2000-01-01 01:00 adbd
srw-rw---- root     inet
2000-01-01 01:00 dnsproxyd
…
…
srw-rw-rw- root     root
2000-01-01 01:00 property_service
srw-rw---- root     radio
2000-01-01 01:00 rild
```

Inside the .rc files you will see a lot of setprop commands, which are used to set various environment variables. In the next section, we will see how the init application uses the property service to inform the boot animation code to start execution after it successfully initializes the graphics stack.

```
on fs
    mount_all /fstab.odroidu
    setprop ro.crypto.fuse_sdcard true

#============================================================
#
# wi-fi post data
#
#============================================================
    mkdir /data/misc/wifi 0770 wifi wifi
    mkdir /data/misc/wifi/sockets 0770 wifi wifi
    mkdir /data/misc/dhcp 0770 dhcp dhcp
    chown dhcp dhcp /data/misc/dhcp

    setprop wifi.interface "wlan0"
    setprop wlan.interface "wlan0"
    setprop wlan.driver.status "ok"

    chown system system /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
    chmod 0666 /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
    chown system system /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
    chmod 0666 /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
    chown system system /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
    chmod 0666 /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq

# Permissions for backlight
    chown system system /sys/class/backlight/pwm-backlight.0/brightness
    chmod 0666 /sys/class/backlight/pwm-backlight.0/brightness

on post-fs-data
    # we will remap this as /mnt/sdcard with the sdcard fuse tool
    mkdir /data/media 0775 media_rw media_rw
    setprop vold.post_fs_data_done 1

# su daemon
service su_daemon /system/xbin/su --daemon
    class core
    user root
    group root
    oneshot
```

Figure 3 - Setprop inside .rc file

Boot Animation

If you have ever owned any kind of Android device, you have seen the familiar Android boot animation when you start up the device. This boot animation is actually an application that is defined inside the .rc files and executed as part of the init process. The boot animation images that you see is actually a sequence of graphics that are played continuously. In this section, we will add a boot animation inside ODROID-U3. The updated boot animation graphics were downloaded from a community website at http://bit.ly/1wRbAL2, and you can select any of the available .zip files on that website.

Installation

The steps that need to be done to install the boot animation are outlined below. Remember that the Android source code must be recompiled in order to see the results. For more information on compiling Android from source for the ODROID-U3, please refer to my previous ODROID magazine article at http://bit.ly/1vkwuYk.

1. Open the file init.odroid.rc and add the following line:

```
service bootanim /system/bin/
bootanimation

    class main
    user graphics
    group graphics
    disabled
    oneshot
```

2. The new service is called bootanim, and it points to the internal application called bootanimation. Notice that the service is marked as disabled. The rationale behind this is that Android will only run bootanimation after the SurfaceFlinger service has successfully initialized, which is a core component for Android graphics. As you can imagine, if the main graphics service of Android did not initialize successfully, that means something is wrong and the graphics will not work when Android starts up.

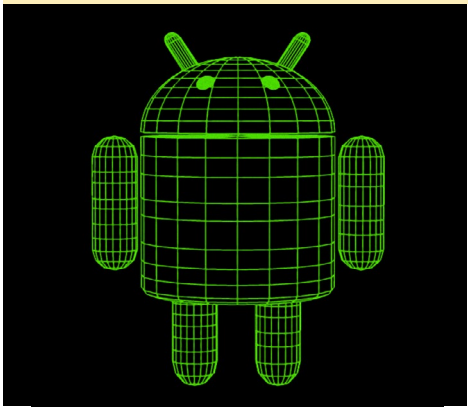
3. The next step is to copy the graphic file for the boot animation to be compiled as part of the building process. The file that the bootanimation application needs must be named bootanimation.zip, and when it is packaged during the build process, it can be found inside the system/media folder as seen in Figure 5.

4. The last step is to inform the build script to copy the bootanimation.zip file to system/media folder during building. This needs to be specified inside the device/hardkernel/odroidu/device.mk file. Add the following command, and compare the script with the one shown in the screenshot.

```
$(LOCAL_PATH)/bootanimation.
zip:system/media/bootanimation.
zip
```

Once you complete the above steps, compile the Android source code and enjoy your new boot animation! For more information on the .rc init language, please visit http://bit.ly/14Cz7sF.

An example of an Android logo animation



```
void SurfaceFlinger::startBootAnim() {
    // start boot animation
    property_set("service.bootanim.exit", "0");
    property_set("ctl.start", "bootanim");
}
```

Figure 4 - Boot animation trigger using property_set

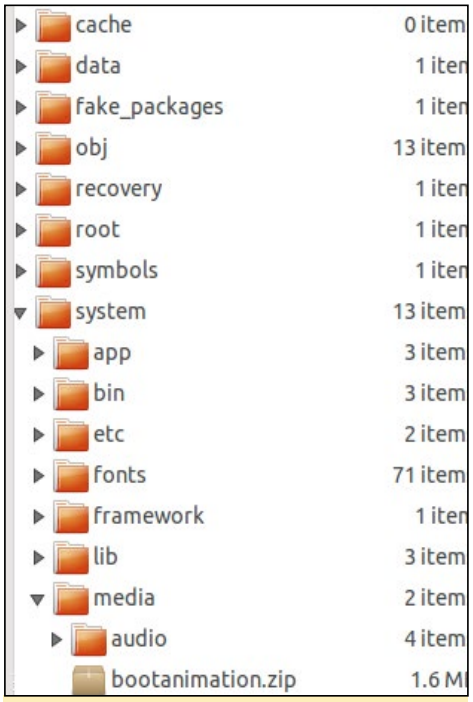


Figure 5 - Inside the system/media folder

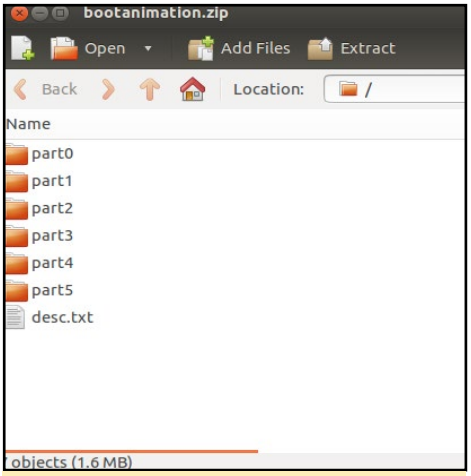


Figure 6 - Inside bootanimation.zip

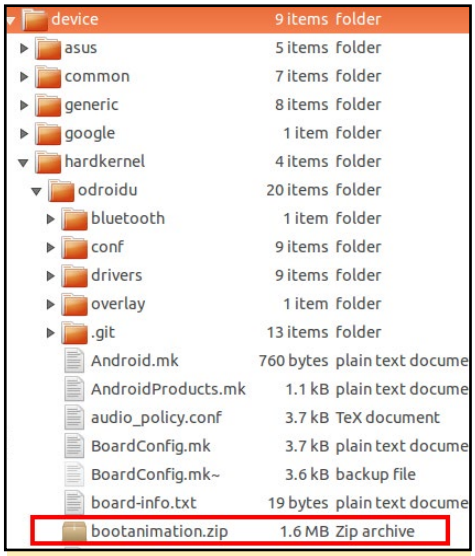


Figure 7 - The bootanimation.zip file must reside inside the device/hardkernel/odroidu directory

This blue plasma boot animation can be automatically run at system startup

Figure 8 - Boot animation copy script

```
# Init files
PRODUCT_COPY_FILES += \
    $(LOCAL_PATH)/conf/init.odroidu.rc:root/init.odroidu.rc \
    $(LOCAL_PATH)/conf/init.odroidu.usb.rc:root/init.odroidu.usb.rc \
    $(LOCAL_PATH)/conf/fstab.odroidu:root/fstab.odroidu \
    $(LOCAL_PATH)/conf/fstab.odroidu.sdboot:root/fstab.odroidu.sdboot \
    $(LOCAL_PATH)/bootanimation.zip:system/media/bootanimation.zip
```

The Android Story

Worldwide Android Market Share

Android, Inc. was founded in Palo Alto, California, United States by Andy Rubin, Rich Miner, Nick Sears and Chris White. — October, 2003

Google acquired Android Inc. — August 2005

The Open Handset Alliance, a consortium of several companies was formed. — 5 November, 2007

Android Beta SDK Released. — 12 November, 2007

0.50 %

The first Android device, the HTC Dream (G1), featuring Android 1.0. — 23 September, 2008

Android 1.0

- Integration with Google Services.
- Web browser to show, zoom and pan full HTML and XHTML web pages, multiple pages show as windows.
- Android Market app downloads and updates.
- Multitasking, Instant Messaging, Wi-Fi and Bluetooth.

Android 1.1 update for Android was released for T-Mobile G1 only. — 9 February, 2009

Based on Linux kernel 2.6.27, the official 1.5 (Cupcake) update for Android was released. — 30 April 2009

Android 1.5 (Cupcake)

- Faster Camera start-up and image capture.
- Much faster acquisition of GPS location (powered by SUPL AGPS).
- On-screen soft keyboard.
- Directly upload videos to YouTube and Picassa.

Based on Linux kernel 2.6.29, the 1.6 (Donut) SDK was released. — 15 September 2009

Android 1.6 (Donut)

- Quick Search Box and Voice Search
- integrated camera, camcorder, and gallery, toggle between still and video capture modes
- Battery usage indicator
- CDMA Support
- Multilingual text-to-speech function

Based on Linux kernel 2.6.29, the 2.0 (Eclair) SDK was released. — 26 October 2009

Android 2.0 (Eclair)

- Multiple accounts for email and contact synchronization.
- Microsoft Exchange Support for syncing of e-mail.
- Bluetooth 2.1 support
- New browser User Interface and support for HTML5.
- New calendar features

MEET AN ODROIDIAN

SURIYAN RAMASAMI: ODROID ENTHUSIAST AND TALENTED COMPUTER HOBBYIST

edited by Rob Roy

Suriyan hiking on the trail as part of a 3 day backpacking trip across the Ohlone Wilderness

Please tell us a little about yourself.

My name is Suriyan Ramasami. I was born in India, and currently live in the San Francisco Bay Area with my wife and daughter. I am 44, and hope to go through my very own mid-life crisis soon! I currently work as a computer engineer for a database company.

How did you get started with computers?

I was lucky enough to have a father that worked for the Indian government, and he was sent to London. I was so excited when he came home with an Atari 130XE! He also brought home many cartridges consisting of video games, the BASIC interpreter and an Assembler. I have been fascinated by computers ever since. The first programming languages that I ever worked on were BASIC and the 6502 assembly language. Oh, those LDA/STA opcodes!

What drew you to the ODROID platform?

I have a monitor, keyboard and mouse at work, and a similar setup at home. I did not want to lug a laptop between home and work, and was looking for a small form factor device which I could carry instead of a laptop. It also had to be powerful enough to be able to handle a GUI desktop. I was considering the Intel NUC, and then I stumbled upon ODROID computers. I still do not use it as my desktop yet, but it did draw me to it.

Celebrating Holi, the Festival of Colors, in India

What inspired you to build an improved boot loader for the ODROID boards?

I was searching for a backup solution for home, and found the GoFlex Home. I wasn't very happy with the solution, and that is when I found a hack on the ArchLinux ARM web site for installing ALARM on it. This introduced me to u-boot, and I loved that it had netconsole, USB access, SATA and Ethernet control inside of u-boot. I was then motivated to add the same support in mainline u-boot. I wanted similar u-boot behavior for the ODROID, and from that came my success in providing USB access from within the ODROID's u-boot application.

Do you have any other boot loader improvement projects in mind?

I have four projects in mind. The first one, is to have HDMI support in U-Boot for the ODROIDs. This would let one interact with the boot loader using an USB keyboard, which is already supported, along with an HDMI monitor. The second one, is to be able to use the USB OTG port as a means of communicating with the boot loader, which achieves the same purpose as my first project. The third project is to be able to use the USB3 ports on the ODROID-XU series from the boot loader. My fourth one is to add support for the ODROID-X model, unifying them with the already existing mainline support, along with the ODROID-U2/U3 and the X2.

Have you worked on other projects related to ODROID's?

I have been involved in getting XEN support for the ODROID-XU upstream. I shall soon be working on getting XEN support for the ODROID-XU3 as well.

What hobbies and interests do you have apart from computers?

I like hikes, long walks, gardening, playing the guitar, and vacationing in different countries. I have been to India, where I was born, the United States, where I live, China (Zhang Ye/ Lanzhou/Beijing), Indonesia (Kawah Putih, Ancol, Pondok Gede, Bandung, Jakarta), Bangkok, Australia, Singapore, Spain (Madrid, Barcelona), France (Nice, Paris), Costa Rica, Mexico and Canada. I also like to participate in many of the South Asian community cultura events, including SEF Dandia and Stanford Holi.

Are you involved with any other projects unrelated to the ODROID?

My work projects allow me to experiment with technologies such as FCoE, Infiniband, Hyperswap and Virtualization.

What type of hardware innovations would you like to see for future Hardkernel boards?

SATA would definitely be high on my list, partly because of my desire to use it as a desktop replacement computer. I do not see this happening with the future Exynos SoCs model, mainly because they are intended to be used as a tablet or smartphone, and a SATA drive isn't usually available for those types of devices. 1 GB ethernet would be nice for interlinking ODROIDs. Both of these improvements would make the hardware more appealing to the server market.

What advice do you have for someone wanting to get started with programming?

I think that in this age, where all information is available under your fingertips, it can get quite overwhelming. Where does one start? My advice would be that it should be need-driven. You should need or want something, and that will give you the motivation to dig far or deep enough to accomplish it. I know that this is a generic statement that can apply to anything!

Suriyan is a true ODROID ninja, with computer skills beyond belief!