

ODROID

Year Six
Issue #61
Jan 2019

Magazine

WE HAVE BEEN DOING THIS FOR A LONG TIME NOW!



Our Anniversary RETROSPECTIVE

WIFI ACCESS POINT:
SBC WIFI CONNECTIVITY
SHARING ON YOUR ODROIDS

SWARM CLUSTER:
A 3D PRINTED I/O RACK MODEL,
PLUS TUTORIAL TO ASSEMBLE





Building Reicast: A Dreamcast emulator for Your ODROID

© January 1, 2019

This guide will go over how to build Reicast from scratch



Remote Control GPIO: Python Version of the Atlas Toolkit

© January 1, 2019

I wanted to control the GPIO of my headless ODROID-C2 with a smartphone, so I developed a web application which can be seen as a web interface to the WiringPi library.



ODROID Magazine Retrospective: A Look at the Evolution of the Magazine over The First 5 Years

© January 1, 2019

Come with us, this is our retrospective for our first five years of our Magazine.



Introducing NEMS Linux – Part 4: NEMS 1.5

© January 1, 2019

With the recent release of NEMS Linux 1.5 I'd like to dial it back a little and introduce you to the most sophisticated out-of-the-box Nagios Enterprise Monitoring Server out there.



Building Kodi: Enabling Automatic Frame Rate Control

© January 1, 2019

This guide assumes that you have installed the packages from the first post, <https://forum.odroid.com/viewtopic.php?f=98&t=32173#p233475>, and are using the exact same OS image, as this is not tested on any other image. We are again modifying the kernel, and I am in no way responsible for any damage on your hardware. ▶



Linux Gaming: PC-Engine / TurboGrafx – Part 1

© January 1, 2019

A little while back, I talked about the PC-Engine / TurboGrafx, and that I think it's an underrated system.



ODROID-XU4 Review: Better Performance Than the Raspberry Pi, Plus USB3 and Gigabit Ethernet For USD\$60

© January 1, 2019

Hardkernel recently sent over the ODROID-XU4 for benchmarking.



BASH Basics: Introduction to BASH – Part 7

© January 1, 2019

After all the theoretical stuff in the last few parts of this series, we're going to take a breather and deal with clever ways to increase productivity in BASH when working with ODROID. The default Ubuntu MATE system is chock-full of useful utilities without requiring so much as installing additional ▶



ODROID-HC1 Swarm Cluster

© January 1, 2019

The Docker team has developed a clustering and scheduling tool for Docker containers, called swarm. This article describes how one can create an ODRROID-HC1 based swarm cluster. This cluster fits in a 19" rack.



Wireless Access Point

© January 1, 2019

Hardkernel offers a few USB based WiFi modules to enable WiFi connectivity for their Single Board Computers (SBCs). Some of them can also be used to deploy these SBCs as Access Points.



Meet An ODRROIDian: Justin Lee, CEO of Hardkernel

© January 1, 2019

It is really hard to remember what happened in the past 5 years due to the limitation of my brain capacity, but let me try to recall what changes have happened. 5 years ago, Hardkernel had 11 people including me but we have 26 members today.

ODROID XU4 Reicast Emulation on VU5A with GBM ...



```
$ strip reicast_libretro.so  
$ cp reicast_libretro.so  
~/.config/retroarch/cores/.
```

First, you will need some BIOS files for NAOMI and Dreamcast, which can be found at <https://www.libretro.com/index.php/reicast-libretro-now-supports-naomi-other-additions/> and <https://docs.libretro.com/library/reicast/>. If you want to know the md5 checksum of the NAOMI BIOS file, you can take a look into the core info file `/home/odroid/.config/retroarch/cores/reicast_libretro.info`

Figure 2 – Reicast Emulator on the ODROID-XU4 <https://www.youtube.com/watch?v=j0jEucQx-vM>

Build from source

To build Reicast from source, type the following commands:

```
$ cd ~  
$ git clone  
https://github.com/libretro/reicast-  
emulator.git  
$ wget -O xu4.patch  
https://pastebin.com/raw/pfVjnVs3  
$ patch -p1 < xu4.patch  
$ platform=odroid ARCH=arm make -j7
```

In order to play games at a decent speed, once you have loaded a game, open the RetroArch menu, go to Core Options, find the following settings and change them to:

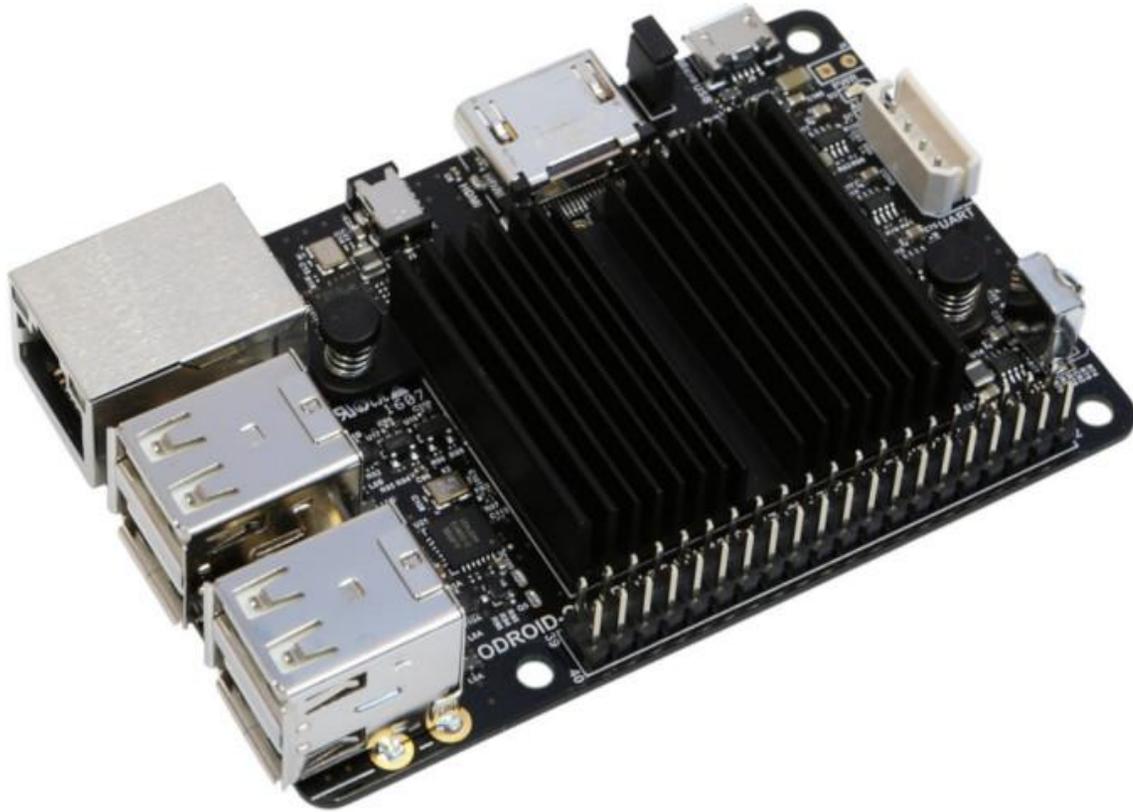
```
reicast_framerate = "normal"  
reicast_enable_rttb = "enabled"  
reicast_threaded_rendering = "enabled"
```

For more information and further posts, please see the original thread at

<https://forum.odroid.com/viewtopic.php?f=98&t=32173#p234251>.

Remote Control GPIO: Python Version of the Atlas Toolkit

© January 1, 2019 By @Epeios ODRROID-C2, Tinkering, Tutorial



I wanted to control the GPIO of my headless ODRROID-C2 with a smartphone, so I developed a web application which can be seen as a web interface to the WiringPi library. In this first release, you can set the pin mode between IN, OUT and (software) PWM, and, for the OUT and PWM mode, set the value of the pin. To use it with the Python version of the WiringPi library installed, simply type the following commands:

```
$ git clone http://github.com/epeios-q37/atlas-python
$ sudo python GPIO
```

After the second command, open the displayed URL in a web browser. From that web page, you can now edit the GPIO status of your ODRROID-C2. The GPIO and RGB applications are designed to be used on an ODRROID-C2. The Python version of WiringPi must be installed using the following command:

```
$ pip install wiringpi
```

The application has to be launched with sudo:

```
$ sudo python GPIO
```

or

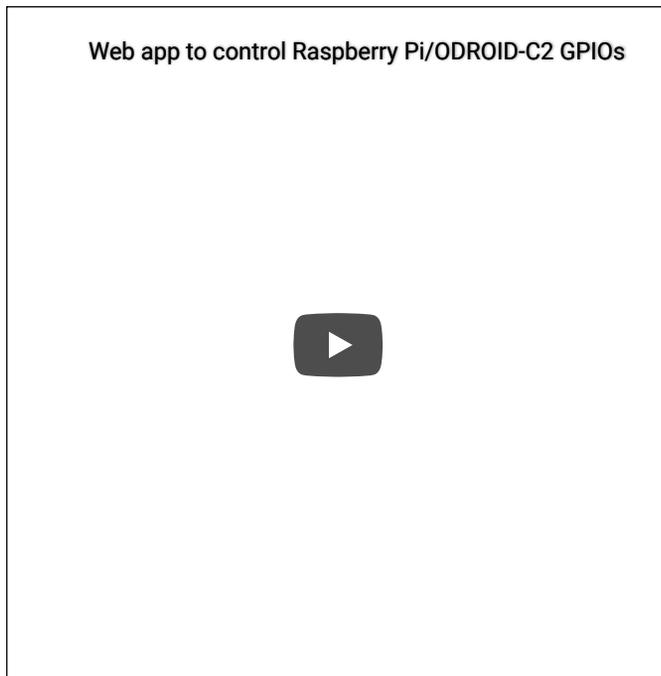
```
$ sudo python RGB
```

The following applications are only examples. You can easily write your own applications.

RGB Led control with Raspberry Pi/ODRROID-C2



(Figure 1 - RGB Application demonstration <https://www.youtube.com/watch?v=4m5CyhT7yiA>)



(Figure 2 - GPIO Application demonstration <https://www.youtube.com/watch?v=I-cVCivRb9o>)

The RGB application is designed to control a RGB led. There is a video at <https://www.youtube.com/watch?v=4m5CyhT7yiA> to demonstrate how it works. The GPIO allows to control the basic pins. There is a video at <https://www.youtube.com/watch?v=I-cVCivRb9o> to demonstrate how it works. For further information, please see the project's git README page at <https://github.com/epeios-q37/atlas-python/blob/master/README.md>.

ODROID Magazine Retrospective: A Look at the Evolution of the Magazine over The First 5 Years

January 1, 2019 By Bruno Doiche Uncategorized



You are reading something that most brewing board groups wish for: a magazine that you can point to your friends to who show interest in those quizzical circuit boards that you fiddle with. Inevitably, when you tell them to go to the forum, as great as any forum can be, it is daunting to grasp where to start exploring it. But, as luck would have it, one day the following post was on the forum:



With this forum post on the ODROID forum in November 2013, things started rolling:



Ubuntu 13.10
Install and troubleshooting (the easy way)

Use C/C++/Python/Java and IDE to:
 Hack your hardware and control LEDs

Start programming right away!

Linux chroot on
 Android • Gaming •
 XBMC Media Center

RPi vs U2
 How do they stand against
 each other?

A few days later we had a cover to get everyone's interest buzzing and we started working together:

So, that was how it began! Where have we been all this time; care to recall with us?

2014 – Year One

Our first year started with the launch of the ODROID-U3, which came to replace the ODROID-U2 and existed alongside the ODROID=X2. The first issue was a great feat, where the magazine team got together online and we got to know each other. For issue number 2, at the height of the Minecraft craze, Hardkernel showed us how to turn the U3 into a Giant tablet, and astonished many users with what was the beginning of our magazine trend of showcasing incredible projects. For the next issue, we presented the thing that has driven most computer users since the dawn of computing: GAMES. It was our first edition where games got the spotlight, and we never tire of it, as you will see. (In fact, if let unchecked, we would make the magazine a games magazine, but then...)



Picture 1 – 2014 January February and March issues

For the next issue, we had a complete change of pace, and learned how to embed an ODROID into a truck, and yes, at this time we couldn't get enough Minecraft in our lives. In May 2014, we showcased the first ODROID-powered robot, and with the release of the ODROID-UPS and the ODROID-SHOW, it never was easier to create new and portable projects. (We also are aware that, since that time, we are being monitored on how we treat our dear robot overlords!) At this point, the readers were asking more from us, so we started covering Operating Systems, and dedicated our June issue to compare a good bunch of prebuilt Ubuntu images for users to look at and decide what was best for their boards. At that time, what rocked our devices' hearts was the venerable Ubuntu 13.04, with 13.10 getting the spotlight, while at the forefront, there were some bleeding edge users running Ubuntu 14.04.



Picture 2 – 2014 April May and June issues

By the next month, we were halfway the year, and for the northern hemisphere summer, Hardkernel surprised the U3 users with a Weather board device, and although the cover does not gave much headline, we were looking at cluster computing, a fan favorite topic. Next, we had the unfortunately short lived ODROID-W, an amazing wearable computer that took all users and let everyone wishing for one. Although short lived, our hearts will always be with you W. You were awesome! In September, we have shown that

our users were going deep into projects, and we showcased a ODROID-U3 powered robot inside a WALL-E toy, and users were turning their ODROIDS into laptops, showing how stable using the Linux on ODROIDS were in just nine months of magazine.



Picture 3 – 2014 July August and September issues

For the Fall issue, we get into October 2014, and a new product comes out of Hardkernel: the venerable ODROID-XU3. It was a powerhouse, with 8 cores and plenty of ports. And while our heads were still spinning around it, alongside it, we got the ODROID SMART POWER to review. For November 2014, we delved into virtualization on the ODROIDS, showing projects that make use of our devices' low power consumption. Not being a tad more modest that we needed to be, we demonstrated an ODROID cluster with 96 cores! In December, we finished our first year with the ODROID-C1, bringing native gigabit for the users, and there were liquid cooled ODROIDS everywhere!



Picture 4 – 2014 October November and December issues

2015 – Year Two

At the beginning of our year 2, we were all head in the clouds, literally. We started our year teaching our users how to better integrate their devices with the whistles and bells that the 2015 cloud computing had to offer. And guess what we were looking at too? If you guessed games, you couldn't be more correct. In

February, we were fiddling more with the Weather Board, but coming from cloud computing, we were all raging to run Docker on our beloved ODROIDS, showing that our devices could be a complex environment unto themselves. Then, after going all tech nerdy, we got back to our project roots and we went airborne by showcasing a Quadcopter all powered by ODROIDS, using Navio+.



Picture 5 – 2015 January February and March issues

April was a great month, as we demonstrated a security focused Linux distribution called Kali, and detailed a DIY light show for your ODROID media center with Ambilight. If you could guess which topic we never tire to talk about, which would be? If it was GAMES, you would be right, since in May 2015, we talked about our hearts' desire to play even more games, this time with a spiffy Nintendo 64 emulation tutorial. Our 2015 mid-year issue had an incredible Desktop Case showcasing (no pun intended) that users look around to create cool form factors for their ODROID desktop builds, and a nod to the original ODROID that started it all.



Picture 6 – 2015 April May and June issues

In July, we always like to remember fondly that we used to take our school out time, to do what? If you guessed games, this time you were not entirely wrong (with games you never are wrong). We did an article on rare gaming, but the spotlight went to watching movies and TV shows using Popcorn Time. In August, the ODROID-XU3 had to step aside as the ODROID-

XU4 came to take the crown as the fastest ARM based computing board from Hardkernel, a title that stands up to this day that I am writing this article! By September we were missing creating new robotic projects and then surprised everyone with a how-to for a Robot Lamp companion. Hardkernel launched the ODROID Cloudshell, which is a cool NAS solution that our users had the best time with.



Picture 7 – 2015 July August and September issues

For October, we went all back to our web roots and taught how to run the Apache Tomcat to give you, our beloved readers, the means to host your own webpages! Then, after doing everything else, Hardkernel presents us with another cool device: the HiFi Shield, giving audiophiles an amazing sound system with the ODROID. And another year goes by, and when you think the community would be done, what do we showcase? A ROBOT GOALKEEPER! If you wouldn't guess that we would go full humanoid with ODROID robots, you were surprisingly wrong!



Picture 8 – 2015 October November and December issues

2016 – Year Three

Our third year started with what? Someone who read the article this far, would wonder if it would be games. But our January 2016 issue was not only speaking for gamers, but instead is speaking for SERIOUS GAMERS, as Hardkernel presented their Universal Motion Joypad! In February 2016, we went portable again,

with the release of the ODROID-C0. We showcased a new robot kit, and dabbled into the now defunct Cyanogenmod for the X/XU line of boards. In March 2016, the all new all powerful ODROID-C2 came to market, giving users that really needed a boost on 64-bit processing projects an all new platform to work with.



Picture 9 – 2016 January February and March issues

April 2016 saw the release of the oCAM, a camera designed to work with your ODROID devices, and we began the augmented reality age on ODROIDS. We usually are fun spirited but serious, yet May 2016 was a very serious issue, pointing all you needed to learn to start delving on the deep web with your ODROID device, as well as understanding Wireless Code Injection. We followed with the June edition presenting a complete guide teaching our readers on how to assemble a Touchscreen Table to work with your ODROID-XU4, along with a great tutorial on how to setup your Samba server.



Picture 10 – 2016 April May and June issues

In July 2016, the community came out with a great optimized build for Minecraft, and we couldn't resist but put it on the cover. After all, it is a game, and if I haven't stressed it enough, we really enjoy games and talking about games! In August, we went for a tour-de-force on the oCAM Global shutter and how you can make the best of this incredible device, starting a trend that we followed for a trimester: talking about the oCAM and security which was as important back

then as it is today. However, suddenly everything changed, and in September 2016, every ODROID user that ran a system compatible with the Android operating system wanted to run only one software: Pokémon GO. Although we didn't yet have a portable case to take our high end ODROIDS to the streets, we weren't detained by our own collective intellects, and worked with the GPS spoofing feature on the Android to go Pokémon hunting in the comfort of our houses!



Picture 11 – 2016 July August and September issues

Then came the fall edition, and we did a tutorial on how to do some wonderful high end gaming using the ODROID-C2 and Gamestream, so you could enjoy even the top PC games in your living room if you wished to do so. On November 2016, we revisited our tutorial for the Ambilight, but now in glorious 4K display, because we can do it! We really love our ODROIDS and think that they are the coolest, and they usually are, unless they are hot, then guess what? We worked out how to cool our ODROIDS, and in December 2016, we showed how to do it with water cooling.



Picture 12 – 2016 October November and December issues

2017 Year Four

In January 2017, we made a special edition with all our ODROID products as a celebration of our products, and also because it was a brand new year, we did a tutorial on how to use ODROIDS to control a Wine

cellar. February followed with the VU8-C, a mobile case for your C class ODROIDS that is the real deal, with a long-lasting battery, a touchscreen, an amazing display, and on top of that, the release of the ODROID SmartPower 2. Then, after a whopping dry spell of four issues without talking about games, we remembered that we kinda like talking about them, and then we presented you, our beloved reader, with a tutorial of how to make your very own ODROID arcade station.



Picture 13 – 2017 January February and March issues

In April 2017, for our 40th issue, we presented Walter: a vintage looking robot powered by ODROID. May 2017 came, and with it, we presented a complete walk-through allowing you to re-purpose that Nintendo 64 case that you found on the flea market in order to play your games in style. Usually the best time of the year for ODROIDians is June, and in 2017 was not different. June 2017 was when the XU4Q was released, coinciding with an excellent article on how to 3D print your own XU4 Case.



Picture 14 – 2017 April May and June issues

For July 2017, we presented an article that didn't rock the world, but it made sure that if the world was rocking, you would know for sure! It was the ODROID Seismograph edition, where we presented how to measure seismic acceleration with the ODROID-C2. In August 2017, we showed how to get started with the world of artificial intelligence programming, and it was a milestone edition, as it was the last edition that

was presented as a pure PDF edition. From then on, we would have the website play a major role on the magazine, changing the way we work on it and how we share articles. The PDF is still available, but the new website allowed users to search, comment, and print their own copy at home. September 2017 saw the introduction of the ODROID-HC1, which enabled you to better create stacks of devices to create your clusters and NAS farms.



Picture 15 – 2017 July August and September issues

October 2017 was all focused on how to build your own Multimedia home server, which is a reader's favorite topic and one of the most popular home projects. LineageOS came to the rescue after the demise of Cyanogenmod for the Android users on ODROIDS, and November 2017 was all dedicated to guiding you in how to compile it on the XU series. To finish 2017, we created a tutorial on operating your ODROID hands-free with the AD2P and the HFP unit on your ODROID-XU4.



Picture 16 – 2017 October November and December issues

2018 Year Five

Five year projects are always a great life strategy, and if you were with us since the beginning, it was time for you to live like a king! We presented the dashboard for ODROID home on January 2018, and in February, the success of the ODROID-HC1 saw seen its successor, the ODROID-HC2, make your clusters even

more powerful than before. March 2018 boasted an incredible mathematics article as we've never seen before, and an impressive article on prime number discovery using the ODROID-C2 graced the pages of our humble magazine.



Picture 17 – 2018 January February and March issues

We always can count on innovative and interesting projects with our devices. April 2018 had an article showing an ODROID-XU4Q being used as a portable sound studio! May 2018 was time to revisit what your ODROID can bring to your car, and in par with the all new software for Android Auto, you could go farther than ever before with it. June was the month for gamers. After all, if you think we would ever get tired of games, you are surely wrong, Hardkernel released the GameStation TURBO, a case for your XU4 resembling the Nintendo 64 case!



Picture 18 – 2018 April May and June issues

And while you were still rejoicing on your GameStation, Hardkernel doubled down and releases the amazing ODROID-GO to celebrate 10 years of Hardkernel, a device that can do wonders to (you guessed right) your gaming life! What a July that was! After all this gaming, in August 2018 we came up with a security-driven issue teaching you to manage all your online passwords with KeePass. September was a nostalgia fest where we published a tutorial to emulate Commodore games, celebrating a computer that we spent all our teenage years loving.



Picture 19 – 2018 July August and September issues

In October 2018, we made another article on how to use your ODROID with your car as a Driving Infotainment System, with an even more mature Android Auto release for the ODROID-C2. Then came November 2018 which gave us news of the first X86 ODROID came to be, the ODROID-H2, ushering a new era that we will certainly enjoy. And finally, December 2018, we ended our first five-year mission with the ODROID-GO turned into a Star Trek tricorder!



Picture 20 – 2018 October November and December issues

You are now reading our milestone 5th year anniversary issue, and we thank you for joining us on this computing journey into the future!

***a big shoutout to Starline on its 5 year anniversary badge that we used on our cover!**

Introducing NEMS Linux – Part 4: NEMS 1.5

January 1, 2019 By Robbie Ferguson Uncategorized



If you've followed along with the "Introducing NEMS Linux" series these past few months, you're already quite familiar with what NEMS Linux means to your network. December's exercise-heavy article almost broke the needle on my nerd-o-meter, so with the recent release of NEMS Linux 1.5 I'd like to dial it back a little and introduce you to the most sophisticated out-of-the-box Nagios Enterprise Monitoring Server out there, available now for the ODROID-XU4 platform. I chose the XU4Q as my main development board for NEMS 1.5, and it was a great board to work with.

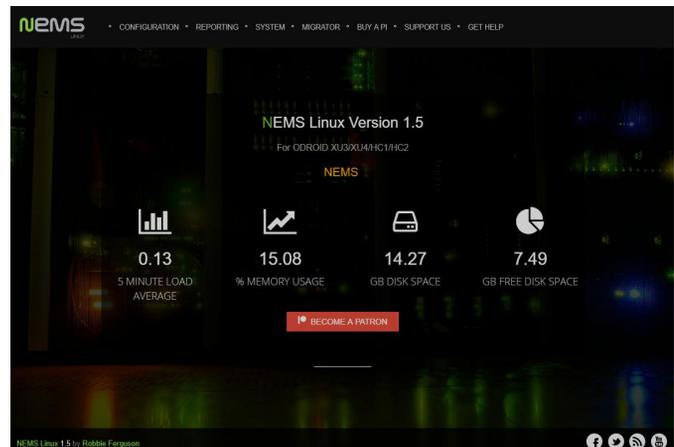


Figure 1 - The NEMS Linux 1.5 dashboard

While this is a quick introduction to some of the standout features of NEMS Linux 1.5, feel free to [browse the changelog](#) to really get a sense for all that has gone into this exciting release. NEMS Linux has always been about bringing an exceptional, feature-rich monitoring server to networks of all sizes. Version 1.5 not only enhances the experience with a suite of improvements, but also begins a focus of bringing NEMS Linux to big business environments. Whether

you're part of a data centre, a school board, a Fortune 500 company, or just a home network, NEMS is built for you. Let's go through some of the enhancements you'll find in NEMS Linux 1.5.

My focus on enterprise doesn't mean these new features aren't also suitable for the small computer shop. Quite the contrary: It means that the smaller network user also benefits from enterprise-grade features. One such feature designed specifically to assist you with the software auditing process is the NEMS System Settings Tool, which now includes the option to change the schedule for automatic software updates, or disable them entirely. The goal is to allow the enterprise sysadmin an opportunity to audit the software. Disabling NEMS' automatic update feature to prevent rollout of unapproved software revisions gives you the chance to test and audit updates before rolling them out to production NEMS servers.

NEMS Update

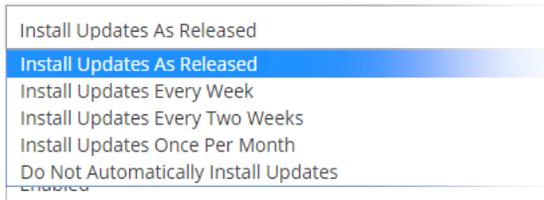


Figure 2 – You can now control NEMS automatic updates

Your NEMS Hardware ID has been moved out of NEMS System Settings Tool and a new feature has been added to the Configuration menu: NEMS Server Overview. This helpful page provides everything you need to know about your NEMS Linux server on a single screen. For example, in NEMS Server Overview, you'll find the LAN IP address of your NEMS Linux server—helpful if `\nems.local` isn't resolving for you. You'll also find the version of NEMS you are using and, of course, your NEMS Hardware ID, amidst several other server-specific items.

| | |
|--------------|-------------------------------------|
| Host Alias | Nagios Enterprise Monitoring Server |
| Service Name | NEMS SBC Temperature |
| Check output | TEMPERATURE OK - CPU Temp: 58.0 °C |

Figure 3 – Monitor your ODRROID-XU4 CPU temperature

One of the goals of NEMS Linux 1.5 has been to introduce several user-requested features as submitted in the community forum. This has meant

integrating a number of new check commands. Along the way I also set up a few new out-of-the-box demo services, including CPU temperature monitoring and Internet speed testing . I've also included Cisco SNMP monitoring, IPMI sensor monitoring, VMWare ESXi hardware monitoring, QNAP device monitoring, MS SQL monitoring, and more.

Service Overview

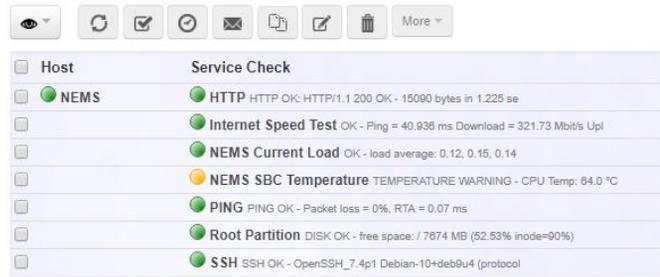


Figure 4 – New NEMS Sample Services in NEMS Linux 1.5

I really wanted to take NEMS notifications to the next level. Nagios notifications are by default very 1990s, but Heini Holm Andersen had already done some amazing work on responsive HTML email templates for Nagios. I reached out to him to discuss licensing his software for use beginning in NEMS Linux 1.5. His products are, by default, not licensed for distribution or modification. I was so pleased to hear back and learn that he loves NEMS Linux, and is excited about my request. He provided blanket permission for me to modify, distribute, use, and include his plugin in NEMS Linux 1.5+. Its source code is now part of NEMS Linux and can be viewed as `nems_sendmail_host` and `nems_sendmail_service` in [NEMS Migrator](#). I adapted the scripts for NEMS Linux so they now pull the SMTP configuration from NEMS System Settings Tool. I also added some customization to the overall look of the emails and added the NEMS Server Alias to each notification, which will help those of you who use multiple NEMS servers—another enterprise-centered upgrade.

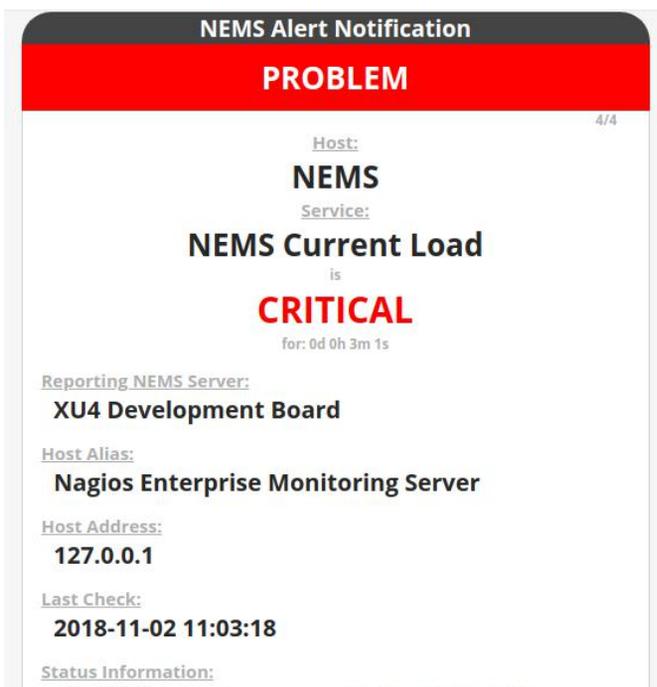


Figure 5 – Beautiful New HTML Email Notifications in NEMS Linux 1.5

Other notification options in NEMS Linux 1.5 include the Pushover app as well as a newly overhauled Telegram notification system that makes setting up a Telegram bot incredibly simple.

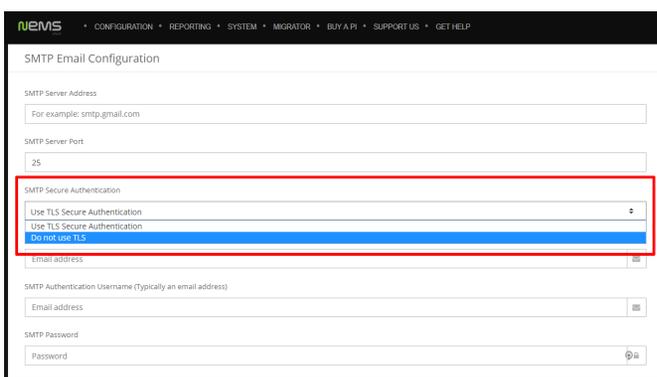


Figure 6 – NEMS SST Now Allows Insecure SMTP Servers

The CSF/LFD firewall now comes pre-configured to protect you from web-based attack, and you can now disable TLS for SMTP (a heavily requested option for sysadmins who use an internal relay). It's also worth noting that the underlying software has been upgraded in the new release. For example, Nagios Core has been upgraded to 4.4.2, and PHP is now version 7.2, which introduces the Sodium cryptography library used by NEMS Cloud Services to encrypt confidential data.

NEMS Linux is absolutely free to download and use. However, I do offer some optional subscription-based services which I'd like to touch on. With a growing list

of optional cloud-based features in NEMS Linux 1.5, NEMS Migrator's Off-Site Backup has been moved to a new platform called NEMS Cloud Services. All Off-Site Backup accounts have been automatically upgraded, and there is no additional cost; just more features. NEMS Cloud Services also adds preliminary cloud functionality in preparation for some exciting features that will be released throughout the NEMS 1.5 release cycle, such as a consolidated NEMS TV Dashboard in the cloud which allows you to see an overview of multiple NEMS Linux servers in real time.

NEMS Cloud Services also includes NEMS Checkin. A number of NEMS users, especially those who are serious about reliability, don't tend to trust SD cards, so it's not uncommon to setup a second, or even a third NEMS server to monitor your main NEMS server. I call this "NEMScaption." This is redundant, but up until NEMS 1.5 has been a great idea for those who want to know if their NEMS server has gone offline. Now, NEMS Checkin (in NEMS SST) allows your NEMS server to check in with NEMS Cloud Services. If your NEMS server stops checking in, NEMS Cloud Services will send you a notification that your NEMS server appears to be offline. As with all NEMS Cloud Services, this useful feature is entirely optional, and is disabled by default.

How to Upgrade to NEMS Linux 1.5

1. The NEMS Migrator tool allows you to export/backup your complete NEMS configuration as well as import a previous backup through its nems-restore tool. This is particularly helpful when upgrading from previous versions of NEMS, or when moving from one platform to another. It saves you having to reconfigure your NEMS deployment just to get the latest features or upgraded hardware. The following steps may be followed to upgrade from any previous version of NEMS to the latest NEMS Linux 1.5. Therefore, if you've already deployed your NEMS Linux 1.4.1 system, you can easily migrate your configuration to NEMS 1.5.
2. Visit your existing NEMS Linux dashboard in your browser and press Migrator and download your local backup.nems file. Alternatively you may access the samba share at \nems.local/ackup (Windows) / smb://nems.local/backup (Linux/macOS) and copy your backup.nems file to your local machine. Set this aside for later importing.

3. Download and burn NEMS Linux 1.5 on a new card. Use a new card so you can always revert back to your existing NEMS Linux 1.4.1 if you have a problem. Boot your device into NEMS Linux 1.5 and give it a few minutes to run its first boot operations, gather some updates, and so on.
4. Initialize NEMS 1.5 as if it was a new deployment by SSH'ing to the NEMS server and typing: `sudo nems-init` Please refer to [Part 1 of this series](#) for the full instructions.
5. Copy the `backup.nems` file you set aside in Step 1 to your NEMS 1.5 deployment's home folder at `\nems.localhome (Windows) / smb://nems.local/home (Linux/macOS)`. SSH into your NEMS server and restore the `backup.nems` file as follows:

```
$ sudo nems-restore ~/backup.nems
```

If you subscribed to NEMS Cloud Services (previously NEMS Migrator Off-Site Backup), replace steps 5 and 6 with:

```
$ sudo nems-restore osb
```

If, on the other hand, you have never used NEMS Linux before and don't need to upgrade from a previous version, you can get up and running by following steps 2-4.

That's all there is to it. You're now up and running with NEMS Linux 1.5. Enjoy!

Learn More

NEMS has an active community forum. I check in quite regularly to provide free support to users. I also offer commercial one-on-one priority support for those needing a higher level of support. NEMS Linux is free to download and use. Its source code is available on GitHub. Download NEMS Linux for ODROID at <https://nemslinux.com/>

You can also follow @NEMSLinux on Twitter or [join us on Discord](#). I've also launched a YouTube channel specifically for NEMS where I will be posting screencasts and tutorials about NEMS Linux. While the content is still to come, I'd really appreciate if you'd subscribe! You'll find the channel [here](#).

About the Author

Robbie Ferguson is the host of Category5 Technology TV and author of NEMS Linux. His TV show is found at <https://category5.tv/> and his blog is <https://baldnerd.com/>.

Building Kodi: Enabling Automatic Frame Rate Control

© January 1, 2019 By @AreaScout ODRROID-XU4, Tutorial



This guide assumes that you have installed the packages from the first post, <https://forum.odroid.com/viewtopic.php?f=98&t=32173#p233475>, and are using the exact same OS image, as this is not tested on any other image. We are again modifying the kernel, and I am in no way responsible for any damage on your hardware. This guide is for tinkering and HIFI/Video hobby enthusiasts that know what they are doing.

Kodi automatic frame rate adjust on ODRROID XU4



Figure 1 – Kodi automatic frame rate adjustment
<https://www.youtube.com/watch?v=mJ0z7qAw2Jo>

Modifying the Kernel

Back in spring of 2018, the great @memeka and myself were about to discover the great unknown. I was studying the DRM mixer source code in the kernel and saw that one plane with NV21 and NV12

color capabilities were disabled, known as the OVERLAY plane. The picture as you see on your TV is made of different layers in DRM that are placed on top of each other, the highest Z position is the top most layer. That OVERLAY plane is normally provided to display videos coming from the hardware decoder of the board which exactly outputs in that NV21/NV12 format but it was disabled! This needed to run DRMPRIME with a zero copy!?

So to sum everything up, the images coming from the hardware decoder have to be scaled down and/or color converted, and the scaler hardware component on the board was not implemented in this kernel version, so it couldn't be used. However, since Kodi makes use of different layers in DRM and needs at least two layers, the concept of the GBM backend was designed. We were thinking about what could be done, and memeka came up with the genius idea to use the cursor plane as the OVERLAY plane and this is what we did first in this howto: patch the kernel in order to swap the disabled OVERLAY plane with the CURSOR plane. Kodi will then be able to switch video resolution and GUI resolution independently!

So, we are not getting DRMPRIME with this, but Kodi with GBM windowing system with great video performance without any tearing, and, as far as I know, automatic frame rate control.

Changing Kodi

Team Kodi has closed the doors to everything other than the Mesa GBM GPU drivers. I hope we will see Mali for Mesa soon to be released, so we have to change that and the use of DRM inside Kodi which means that this will be the second part of this guide: patching Kodi.

Patching the Kernel

Get the source code if you don't have it already:

```
$ git clone
https://github.com/hardkernel/linux.git
$ cd linux
$ wget -O drm_plane_fix.patch
https://pastebin.com/raw/MiTbZsPs
$ patch -p1 < drm_plane_fix.patch

$ make odroidxu4_defconfig
```

```
$ make -j7
$ sudo cp arch/arm/boot/zImage /media/boot/.
$ sudo cp arch/arm/boot/dts/exynos5422-
odroidxu4.dtb /media/boot/.
$ sudo make modules_install
```

Patch Kodi

Use the following commands to patch Kodi:

```
$ git clone https://github.com/xbmc/xbmc.git
$ cd xbmc
$ wget -O kodi.patch
https://pastebin.com/raw/mzAiCU2Q
$ patch -p1 < kodi.patch
```

Patching internal FFmpeg to prevent a memory leak (separate because maybe not needed in the future)

```
$ wget -O FFmpeg_fix.patch
https://pastebin.com/raw/rRbXTfra
$ patch -p1 < FFmpeg_fix.patch Next, you'll
need to get a ton of pre-requirements: $ sudo
apt-get install antlr autoconf automake
autopoint autotools-dev ca-certificates-java
curl debhelper default-jre default-jre-
headless default-libmysqlclient-dev dh-
autoreconf dh-python dh-strip-nondeterminism
doxygen fontforge-common fontforge-nox fonts-
droid-fallback gawk gettext gettext-base
gir1.2-harfbuzz-0.0 gir1.2-ibus-1.0 gperf
groovy icu-devtools intltool-debian ivy java-
common junit4 libantlr-java libao-common
libao-dev libao4 libapache-pom-java
libarchive-zip-perl libasan3 libasm-java
libass-dev libass9 libasyncns0 libavahi-
client-dev libavahi-common-dev libbluetooth-
dev libbluray-dev libbluray2 libboost-
atomic1.65-dev libboost-atomic1.65.1 libboost-
chrono1.65-dev libboost-chrono1.65.1 libboost-
date-time1.65-dev libboost-date-time1.65.1
libboost-dev libboost-filesystem1.65.1
libboost-serialization1.65-dev libboost-
serialization1.65.1 libboost-system1.65-dev
libboost-system1.65.1 libboost-thread-dev
libboost-thread1.65-dev libboost-thread1.65.1
libboost1.65-dev libbs2b0 libbsf-java libbz2-
dev libcap-dev libcapnp-0.6.1 libcdio-dev
libcdio17 libcec-dev libcec4 libchromaprint1
libclang1-6.0 libcommons-cli-java libcommons-
lang-java libcommons-logging-java libcommons-
parent-java libcrossguid-dev libcrossguid0
libcurl4-gnutls-dev libcwiid-dev libcwiid1
libdbus-1-dev libevent-2.1-6 libexpat1-dev
```

```

libfftw3-double3 libfile-stripnondeterminism-
perl libflac8 libflite1 libfluidsynth1
libfontconfig1-dev libfontforge2 libfribidi-
dev libfstrcmp-dev libfstrcmp0 libgcrypt20-dev
libgdraw5 libgif-dev libgif7 libglib2.0-bin
libglib2.0-dev libglib2.0-dev-bin libglvnd-
core-dev libglvnd-dev libgme0 libgmp-dev
libgmpxx4ldbl libgnutls-dane0 libgnutls-
openssl27 libgnutls28-dev libgnutlsxx28
libgpg-error-dev libgraphite2-dev libgsm1
libgtest-dev libhamcrest-java libharfbuzz-dev
libharfbuzz-gobject0 libharfbuzz-icu0
libhawtjni-runtime-java libibus-1.0-5 libibus-
1.0-dev libicu-dev libicu-le-hb-dev libicu-le-
hb0 libiculx60 libidn2-0-dev libidn2-dev
libiso9660-10 libiso9660-dev libjack-jackd2-0
libjansi-java libjansi-native-java libjline2-
java libjpeg-dev libjpeg-turbo8-dev libjpeg8-
dev liblcms2-dev libldb1 libllvm6.0 libltdl-
dev liblzo2-dev libmad0 libmicrohttpd-dev
libmicrohttpd12 libmirclient-dev libmirclient9
libmircommon-dev libmircommon7 libmircookie-
dev libmircookie2 libmircore-dev libmircore1
libmirprotobuf3 libmodplug-dev libmodplug1
libmp3lame0 libmpeg123-0 libmysofa0
libmysqlclient-dev libmysqlclient20 libnfs-dev
libnfs11 libnorm1 libnspr4 libnss3 libogg0
libomxil-bellagio-dev libomxil-bellagio0
libopengl0 libopenjp2-7 libopenmpt0 libopus0
libp11-kit-dev libp8-platform-dev libp8-
platform2 libpangoxft-1.0-0 libpcre16-3
libpcre3-dev libpcre32-3 libpcrecpp0v5 libpgm-
5.2-0 libplist-dev libplist3 libpostproc-dev
libpostproc54 libprotobuf-dev libprotobuf-
lite10 libprotobuf10 libpython-dev
libpython2.7 libpython2.7-dev libqdox-java
librsvg2-bin librubberband2 libsensors4
libservlet3.1-java libshairport-dev
libshairport2 libshine3 lib smbclient
lib smbclient-dev libsnappy1v5 libsndfile1
libsndio-dev libsndio6.1 libsodium23 libsoxr0
libspeex1 libspiro0 libsqlite3-dev libssh-
gcrypt-4 libssh-gcrypt-dev libtag1-dev
libtag1v5 libtag1v5-vanilla libtalloc2
libtasn1-6-dev libtdb1 libtevent0 libtheora0
libtimedate-perl libtinymce-dev
libtinymce2.6.2v5 libtool libtwolame0
libunbound2 libuninameslist1 libvdpau1
libvorbis0a libvorbisenc2 libvorbisfile3
libvpx5 libwavpack1 libwbclient0 libwebpdemux2
libwebpmux3 libx264-152 libx265-146
libxapian30 libxml2-dev libxmu-dev libxmu-
headers libxmu6 libxpp3-java libxvidcore4

```

```

libyajl-dev libyajl2 libzmq5 libzvbi-common
libzvbi0 mysql-common nettle-dev openjdk-11-
jre openjdk-11-jre-headless po-debconf python-
dev python-pil python-talloc python2.7-dev
python3-distutils python3-lib2to3 samba-libs
swig swig3.0 unzip uuid-dev yasm zip ccache
libinput-dev libpulse-dev pulseaudio

```

Generate the Makefile and export some extra compiler switches:

```

$ export CPPFLAGS="-mfloat-abi=hard -marm -
mtune=cortex-a15.cortex-a7 -mcpu=cortex-a15 -
mfpu=neon-vfpv4 -ftree-vectorize -mvectorize-
with-neon-quad -ffast-math" $ export
CXXFLAGS=$CPPFLAGS $ export
CXX_FLAGS=$CXXFLAGS $ mkdir kodi $ cd kodi $
cmake -DCMAKE_BUILD_TYPE=Release -
DCMAKE_INSTALL_PREFIX=/usr/local -
DCORE_PLATFORM_NAME=gbm -
DGBM_RENDER_SYSTEM=gles -DENABLE_ALSA=ON -
DENABLE_AIRTUNES=ON -DENABLE_UPNP=ON -
DENABLE_INTERNAL_FMT=ON -
DENABLE_INTERNAL_RapidJSON=ON -
DENABLE_OPENGL=ON -DENABLE_OPENGL=OFF -
DENABLE_X=OFF -DVERBOSE=OFF -
DENABLE_SMBCLIENT=OFF -DENABLE_CEC=ON -
DENABLE_NEON=ON -DWITH_CPU=armv7-a -
DENABLE_AVAHI=ON -DENABLE_PULSEAUDIO=ON -
DENABLE_CCACHE=ON -DENABLE_APP_AUTONAME=OFF -
DENABLE_DVDCSS=OFF -
DENABLE_INTERNAL_CROSSGUID=OFF -
DENABLE_OPTICAL=OFF -DENABLE_EVENTCLIENTS=ON
-DENABLE_VAAPI=OFF -DENABLE_VDPAU=OFF -
DENABLE_INTERNAL_FLATBUFFERS=ON ../. $ make -
j7 $ sudo make install

```

Run Kodi, then go to Settings -> Player and adjust setting level on the lower left side, there is Basic, Standard, Advanced and Expert chose either Advanced or Expert. Go to Settings -> Player -> Videos and enable adjust display refresh rate to On start/stop. Go to Settings -> System -> Display -> Whitelist and add all resolutions you want to be able to switch automatically.

Fix CEC

With Kernel 4.10+, a new Linux CEC Framework was introduced, so you need a special libcec library. Here is how to build it from source:

Pulse eight platform support library:

```
$ sudo apt-get install cmake build-essential
$ git clone https://github.com/Pulse-
Eight/platform.git
$ mkdir platform/build
$ cd platform/build
$ cmake ..
$ make -j7
$ sudo make install
```

Pulse eight libcec library with new Linux framework support patch. Note, you should uninstall any old libcec files before testing:

```
$ sudo apt-get install cmake libudev-dev
python-dev swig
$ git clone https://github.com/Pulse-
```

```
Eight/libcec.git
$ cd libcec
$ wget -O libcec.patch $
https://github.com/Kwiboo/libcec/commit/48255b
7d4e1cba1050b8abfbd03be37c0737e832.patch
$ git apply libcec.patch
$ mkdir build
$ cd build
$ cmake -DHAVE_LINUX_API=1 ..
$ make -j7
$ sudo make install
$ sudo ldconfig
```

For more information, please see the ODROID forum thread at <https://forum.odroid.com/viewtopic.php?f=98&t=32173#p235249>.

Linux Gaming: PC-Engine / TurboGrafx – Part 1

© January 1, 2019 By Tobias Schaaf Gaming, ODROID-XU4



A little while back, I talked about the PC-Engine / TurboGrafx, and that I think it's an underrated system. I have a lot of things that I normally work on, so time is always too short, and I couldn't play on the PC-Engine as much as I wanted to. Therefore I decided to make a series about this interesting gaming console and write down my experience.

As the system has A LOT of games. I decided to concentrate on the CD based games first, as it's a smaller number and decided I want to look at each and every one of them I have for a short while and decide if I like them or not. For this I will divide them in three sections: games I liked, acceptable games, and games I didn't like.

How I will test these games

I ran these games on my ODROID GameStation Turbo image on an XU3, but it should run perfectly fine on any ODROID SBC. I'm running all of these games in .chd format, as this saves quite some space as well as

preserving audio quality. The latest versions of mednafen_pce_fast libretro core supports this format and I got quite fond of it and convert most of my games into that format where it's supported.

I'll test a couple of games for each part of the series trying to do this in an alphabetic order. I will try and keep playing each game for a time and not only look at the first minute of the game, as some games get better later in the levels. I may skip some games if they exists in both jap. and eng., or even name them double if I forgot that I played it in another language already.

I will try to give a short description of these games and why I like them or not. The games that I find most impressive I will add pictures to it, others may just have a text description. Obviously this is a personal preference and my taste in games may not reflect your own.

Please note, that most of the games on this console are shooters, so if you like shooters, this is your go-to

console for sure. If not, I hope there are enough other games to keep you interested anyway. Once again, I only focus on the CD based games at this time. Many of these games also came out as HuCards / TurboChips and some of these games were even better than their CD counterparts. If you like what you hear and see here I encourage you to try out the HuCard / TurboChip games as well, as there are even more great titles to be found for the system.

Games I liked

4 in 1 Super CD

This game collection comes with Bonk as a mascot. On the start screen you will find 3 games:

Gate of Thunder, Bonk's Adventure and Bonk's Revenge, which is a "hidden" fourth game is Bomberman for this game on the game select menu you press up, right, down, left and button 2, and you will get to Bomberman. This compilation is just perfect, and not a single game on this list is bad.

Bomberman is the original one, and you can either play it on your own or with up to 5 players in total, there is no doubt about it, this game is awesome. Gate of Thunder is the first, but definitely not the last, shooter on this list. It comes with a very nicely animated intro, which is quite long, but can be skipped. PCE CD games often come with long intros or cut-scenes which is actually quite nice, as very few other consoles in that era offered this as a feature.

As with many PC Engine games, the colors are bright compared to the darker color pallet as for example used on the Sega Mega drive. The game comes with nice parallax scrolling and is overall very fun to play. Bonk's Adventure and Bonk's Revenge shouldn't need to be explained. Both are platformers, and Bonk's is somewhat the mascot for the system. You jump, you hit enemies with your head or jump on them, you can eat items to replenish health, or to get special powers like from meat where he will go into rampage mode and either cause a tiny earthquake that can stun enemies, or you just run through them all. These games are pretty awesome and you should really try them out.

The 4 in 1 Super CD is a real gem for the system 4 incredible games all on once CD. This really shows off how good the console can be and was definitely worth its money. If you want to know what this console is all about, I recommend that you try this compilation!

One warning though: if you use save states, please note that the states are always for the CD, not for the game you play. If you play Gate of Thunder and make a save state and then switch to Bomberman and load the save state, you're back in Gate of Thunder. The same situation happens if you switch to Bomberman and save your game (using a save state): you will overwrite your save state of Game of Thunder. Please be aware of that and make sure you select different save states for each game or only play one game at a time with save states.

Akumajou Dracula X - Chi no Rondo / Castlevania: Dracula X (Rondo of Blood)



Figure 1 - Running away from a beast grabbing the key to free Maria



Figure 2- Freeing Maria the second playable character if you got the key

This is one of the best (if not the best) game for the system. This game is amazing. The graphics, sounds, and music is perfect, and so are the controls. Everything is spot on and reacts as it should be. Also this version is a PC Engine exclusive and is only playable as part of the a remake on the PSP.

If you want to see a very nice review of this game check out the YouTube video at <https://www.youtube.com/watch?v=qsVHqoS1Y58>. It describes the game best, in my opinion, and you can see what the game looks like in real action.

Avenger

This shooter starts with an intro that is made out of still pictures and animated sprites, so it's not a video, and has very little in matter of cutscenes going on. The in-game graphics are not very impressive either. There isn't any parallax scrolling going on and enemies are not extremely detailed. Still, the game is rather addicting. It's also very hard, especially in the first level. Before each mission, you can select your weapon, sub weapon and special weapon, where you start with a fast but weak vulcan cannon or a rocket launcher which does a lot more damage, but it's harder to hit things and is a lot slower.

In game, you can collect upgrades to increase your main and sub weapon, and over time you'll get more weapons to choose from. If you update a weapon, the level of the weapon will be taken over to the next mission. The primary weapon can be updated four

times, the secondary weapon twice (if I remember correctly). The special weapon is you usual "bomb" where you have an limited amount per level and which vary in usefulness. Although very hard, it is very addicting, as I said, and you should train your save state and load state key combos to get a little extra help. The music is fine, but the sound can be a little bit annoying especially with the laser in the later levels. Imagine a high pitched sound for each shot you give and the laser firing at 10 shots a second (and yes, each shot makes the sound). Anyway, I rather enjoyed playing the game and looking forward to beating it in the near future.

Bakushou Yoshimoto Shinkigeki

Wow, I don't even know what I'm playing here. This game is all crazy. I have no clue what I'm doing: I jump on people's heads, have a few path decisions, ride rollercoasters, dance with girls, play rock paper scissor, and keep jumping on people's heads. The game is just crazy but also very funny. The graphics are nice with huge bobble head characters. It's so funny, and I love it.

Bonk III - Bonk's Big Adventure



Figure 3 - Huge Bonk, cause being a tiny caveman sometimes isn't enough



Figure 4 – Nice and beautiful drawn levels in Bonks III

This game is more of the Bonk's goodness. This version is pretty much the same as the HuCard version but with better music. In some of the previous versions, the graphics and gameplay were actually changed for the CD version and made the games in fact inferior to their HuCard counterparts (except for the music). So this is a nice to have on as a CD. This time you can play with two players if you want, and next to the regular meat that give you different abilities, you now have blue and red candy that either makes you huge or very tiny depending on which one you eat. This allows you to reach areas that were previously unreachable for you. It's a very fun and addicting game, and you can understand why he's somewhat of a mascot for the system.

Camp California



Figure 5 – The later levels have more details then when you start on the strand



Figure 6 – Freeing one of your captured buddies

I was actually surprised when I played the game. On the first look it was very childish and the graphics not very spectacular, especially the cheesy intro, which looked and felt like it was made for 5 years old, but it turned out the game was actually not that bad at all. You play a bunch of bears and birds and you want to protect your beach from being turned into a nuclear power plant, and for this you have to play music (obviously, right? Cause that is how you drive away these billion dollar companies?). Anyway your friends get kidnapped, your 70s flower power Volkswagen bus get its tire stolen, so you're pretty much screwed. There's only one thing to do: go get your tires back, find your musical instruments and free your friends! This is an action platformer, where you jump, kick, box, or just run enemies over with your skateboard.

The concept is rather simple: the graphics vary from uninteresting plain to nicely detailed and cartoony. There is no parallax scrolling. You have an overview map where you can drive from place to place (after you got your tires back) and different places you can visit. Some of the levels are rather big and you have to find items collect garbage (for recycling as full for your bus) and find and free your friends.

Once you've freed one of your friends, you can play him as a new character, and each one has special abilities, like for example skateboarding for faster travel and higher jumps. It's a rather decent platformer with enough places to explore to keep you busy. The jumping needs some time to get used to, as often you only can move half an inch from one

platform to the next, but tap the controller a tiny bit too long and your character moves at least an inch instead.

The most fascinating thing that I liked was the music of the game which is actually really good. In fact, you'll find songs of the Beach Boys as game music with vocals and all, which was really surprising for me.

Chiki Chiki Boys



Figure 7 - There are a lot of boss fights in Chiki Chiki Boys



Figure 8 - Graphics are colorful and very diverse

This action platformer is very fun to play. It also supports two player cooperative play in case you want to venture out with a friend. The graphics are nice and cartoony, and the sound effects are good as well. It has good music, although there is a lot of it is mostly in the background, and you won't really notice it. The character controls are rather nice, and jumps go easy

and the attacks work well. You can have up to 3 special attacks which can be triggered by pressing "select". These are normally screen filling and do a lot of damage. They also make you invincible for a short while, so it is often helpful when you fight one of the many bosses. You can collect a couple of different items, for example the blue and red pills, which will restore your health either partly or to the full. As the game is completely in Japanese, I have no clue what the story is all about, but I don't think you need it to play this game. The game is quite fun and I really enjoy playing it.

Choujikuu Yousai Macross 2036

This is the first real shooter that I reviewed for this article (which is quite surprising considering how many games I played just to get here). It's a side-scrolling action shooter in the Macross universe. Unfortunately, you can not morph your aircraft at will, but instead you're being morphed depending on the scene you're in. It's a very fast-paced action shooter, but not unfair in doing so. You can upgrade your main weapon by collection power ups, and you can really see your attack getting bigger and stronger.

Killing enemies and destroying objects will earn you points or experience, which can be used at a beginning of a level to buy special weapon to aid you. There are a lot of weapons to choose from and they cost more and more points. At least in easy mode, you'll keep all your power ups when you die and also get a new life every 100,000 points. There are also hidden objects that flash when you shoot them. When you shoot them often enough, they become visible and can also provide extra lives when collected. There are a lot of cutscenes, not only in between missions but sometimes also while on a mission, so that's actually quite nice. Although everything is once again in Japanese, only I still rather enjoyed this game.

Choujikuu Yousai Macross - Eien no Love Song

This one I had a hard time deciding if the lack of understanding is stopping me from enjoying the game or not, since the game and all its cut scenes are in Japanese, which makes it rather hard to understand. Still, I quickly figured out the most important commands and was able to play the game

without missing too much of it I guess. This game is a tactical strategy game similar to history line 1914-1918 or Battle Island games, or as a more recent example, Advanced Wars on the GBA.

True to the Macross universe, you're guiding your pilots through the levels, and can morph them between an airplane for faster travel but lower damage and armor, a mecha that has a lot of armor and does deal more damage, and something in between. If you like tactical strategy games, this game is very nice. It has nice cut-scenes between missions, a nice soundtrack, and good fighting animations.

Cosmic Fantasy II

Do you know this situation where you just want to test something for a little while and suddenly it's a couple of hours later? That's what happened when I first played this game. Cosmic Fantasy II is an RPG game in the style of Fantasy Star or Dragon Quest. The overall graphics are not that impressive, but it makes up for it with really nice cutscenes which are (like the rest of the game) fully in English, so I actually understand what's going on, and I like it. You travel on a world map with random encounters and can visit towns and dungeons. Leveling up can make you a lot stronger, especially at the start. It improves your strength and health and later magic points as well. Since RPGs are not very common on the PC Engine / TurboGrafx this is a nice addition to your library. If you can look past the plain graphics, it's actually a really nice game and I very much enjoyed playing it.

Cotton - Fantastic Night Dreams



Figure 9 - Nice graphics with parallax scrolling



Figure 10 - Keep your most powerful attacks for the boss fights

Cotton should be rather well known shooter game series where you play as a little witch with a fairy familiar. I just hope I don't need to explain this game to anyone. This one is very close to the arcade version. The graphics are great, and the music is, thanks to the CD remix, even better than the arcade version. The game is quite hard, but very fun to play, and the little cutscenes between levels are rather funny. It's definitely worth checking out.

Acceptable games

Advanced Variable Geo

This is one of the few fighters on the game. This game is similar to Street Fighter but with girls only. It's not a bad game, and the sounds and music are fine, but the console doesn't have very much RAM and the sprites

for fighting are rather limited because of this, which makes the moves look very immature and sluggish. The game has nice colors, as most PCE games have, but no parallax scrolling or special effects that would make the game stand out. The only thing in my opinion that redeems the game a little is the support for more than just 2 buttons (as it's usually the norm with PCE games). This is one of the few games where you can use the X and Y button on your gamepad as well, which is logical for fighters, but still rather rare for PCE games. I don't like the game that much.

Ai Chou Aniki

Well this game is "different". It looks like you play a gay bodybuilder, but to be honest, I can't really say, since the game definitely doesn't take itself serious. The graphics are good with different level of parallax scrolling (but not in all levels). You "fly around" as a male bodybuilder, with nothing on except for your underpants and you shoot "bubbles" on... well, let's call them "enemies". Sometimes you even shoot "sparkles" or when your bald head is fully shiny, you can make a huge attack beam that attacks everything in its path. Your second button makes yourself spinning in a pirouette which makes you untouchable, and you can move over the screen. It's an easy way to avoid being attacked.

You cannot aim, it's always selecting the enemy/object closest to you, if you want it or not, but I guess that's part of the game mechanics, as it often means you're not shooting the boss you probably want to shoot. Still, the game is "fun" in a weird and creepy way, but still fun. The music is quite good, but the sound effects get boring really quick. If you are into weird gay bodybuilder shoot em ups, you should try this game.

Ane-San

This is a fighting game for the PC Engine, and although completely in Japanese, it is rather easy to understand. It plays similar to games like Double Dragon or Streets of Rage but with an all female cast, means you only have female characters to choose from. The characters are quite big, with a decent amount of sprites for the animations, but this is probably the reason why I've never seen more than

three enemies at a time on the screen. Gameplay wise it's very repetitive; you're beating up the same enemies over and over again, only changing in color (also normally women), and at the end of a stage you encounter a boss fight. It's actually a mid boss, and then you get to the end boss after some more beating of henchmen. In between stages you can save, do bonus games, buy clothes for your characters and that kind of thing. It's nice to look at, although not very spectacular, but still nice for passing time. In my opinion, it's one of the better beat 'em ups for the console.

Asuka 120% Maxima Burning Fest

This is another fighting game in the likes of Street Fighter, but once again with all girls, in fact school girls, and in a school. The music is very good, although the sound could be better. The graphics are nice for the PC Engine although not impressive. The characters have enough sprites during fights to make it feel fluent, but not enough to look fluent. It's not bad, and although the game only uses 2 buttons, special attacks are rather easy to pull off. The game also exists for the PS1 and Sega Saturn where it looks and plays much better, especially on the Sega Saturn.

Beyond Shadowgate

This is a kind of action adventure game and probably rather rare to find on consoles. It plays in a medieval setting, where the king was murdered and you, his son, got imprisoned, and you have to find/fight your way out of the dungeon. This game reminds me a lot of "Lure of the Temptress" if anyone heard of this game.

This one triumphs through its voice acting, as important scenes are played in CD Quality sound samples. Graphics are not bad, but also not spectacular. Character sprites are big with good level of animation. The music and sound effects are excellent, yet the gameplay is rather hard. In the first couple of minutes, I died numerous times, and it's hard to tell if I'm just doing something wrong, or if I just suck at fighting. Generally this game is very interesting, and I probably look closer into it when I have more time.

Black Hole Assault

The game is a mecha fighting game in a Street Fighter style. The fights are not very good, and I could only figure out a few combos, and most of the time you don't have the time to wait for your combo meter to charge up. The fights are rather quick and sometimes can last less than 10 seconds. In fact, I beat the game in less than half an hour, and don't think it has much of a replay value. There's even a tournament mode or exhibition mode, which doesn't add to the replay value in my opinion.

The best part of the game is the intro and cutscenes, which are subtitled in Japanese, but voiced in English. They are all animated sprites, which look quite good, but when I first saw the game I expected something completely different (more like a shooter). It's not too bad, but it was over quickly.

Blood Gear

The game is a mix of mecha action platformer and RPG game. In fact, the mixture is quite nice. Even the part where you walk around in your power suit and kill enemies is good, especially in the later stages. You kill enemies and they drop "health" kits or weapons, of which you can use up to five of them with your power suit. This makes the game rather interesting. You can shop for items to increase your mecha stats and things like that.

What I like about this game is the art style of the cutscenes and the character portraits (which you have a lot of), the music, sound, the fact that there is a lot of voice acting, and graphics overall are rather nice. What I don't like about this game is that as much as I like the cut-scenes there is no way to skip them. If you've restarted a game or started from an earlier point in the game again, you can't skip the scenes you've already seen.

What I hate about this game is that it's completely in Japanese, and I don't understand a single thing that is being said or written. I bet I would have loved this game if it would have been in English or German, but it does not exist in any other language.

Bomberman '94 Special Version

This is a pre-release, since the final version came out on HuCard (Turbo Chip) instead of CD, so this version

shows what the CD version would have probably looked like. The graphics and sounds are fantastic, and it's really fun to play, yet it's completely in Japanese and offers no single player mode, but you can play it with a total of 5 people. It's sad that this game is not complete, since the sound (and especially the music) is much better on the CD version than on the HuCard version.

Bomberman Panic Bomber

Despite the name, it does not have much in common with Bomberman, and is more of a puzzle game in the style of Puyo Puyo and alike. You collect 3 or more Bomberman heads of the same color in a vertical or horizontal row and they pop. Additionally, there are bombs in the mix, and occasionally a red bomb drops that, if strategically placed, can blow up all the normal bombs. The graphics and sounds are all good, but I'm just not really a fan of these games.

Bonanza Bros

In this game, you play as gangster brothers that break into buildings to make a heist and steal some valuables. You are equipped with a pea shooter that can stun enemies, and you can jump. You need to avoid or disable the guards, grab the valuables and then head to the exit. Every now and then, there is a bonus stage where you have to collect money backs without being "spotted", quite literally, as there are spotlights that you need to avoid. It's a fun little action puzzle game that you can even play with a friend (or your brother). The graphics, sound, and music are ok, but not groundbreaking. It's nice for casual gamers.

Buster Bros.

There isn't much to say about this game. Have you ever heard of "Pang"? It's a game where you shoot bubbles/balls that split into two smaller bubbles/balls until the really tiny ones finally disappear. That's this game: it's just a port of Pang under a different name, but with the same graphics and everything. It's not bad and not awesomely good, just good old Pang with CD music.

Cardangels

This is some kind of strip poker (well not only poker) with anime girls, but you won't see much besides girls

in underwear. Aside from that, there are some decent card games you can play (if you know them). It's nothing special, but is ok to pass some time.

Championship Rally

I was surprised to find a racing game on the system, and even more surprised that it was a game in the style of Micro Machines or Dust Racer 2. It looks nice enough for the system and is really fast, but it seems that I am very bad at it and bump into things over and over again. It's just not my type of game, but really not that bad if you have good reflexes.

Chou Aniki

This is from the same guys that brought us Ai Chou Aniki, but this time it's more of a regular shooter, and they share some of the same assets. The game is fundamentally different though. It's not quite as funny as the other one was, and the shooter aspect is much more serious now, with weapon ups and special attacks (bombs). You can even select your movement speed by pressing the "select" button. There are a lot of bosses throughout the levels, and some "harder" enemies that take more hits than your other enemies. These often drop orange and blue striped pills which act as a power up when you have collected enough of them. You can also have two "familiar" with you, like spheres in other games that are next to you and help you with your shooting. It's not bad, but also not very special.

Color Wars

Color Wars is a puzzle game very similar to Reversi, but with more depth and 4 players at a time. You can place beads of your color on a playing field of 4 by 4 means, 4 beads next to other and above each other, can be placed. You can either place a bead on a empty field, or on a bead of an enemy player, but not on your own beads. Beads of a different color between your beads will turn to your color (also in all directions), and the player which has the most beads on the fields at the end wins. It's a simple but interesting concept, and although there is not much "action" going on, it can be rather nice to play it for a while. The nice relaxing music helps you to plan your next move. Overall the game is nice, but it might not be for everyone.

Games I disliked

The Addams Family

This game is weird, but not in a good way. As an intro, you get a still standing picture and some voice recording where you're told you should find the family treasure vault and get anything you want from there, but they try to hinder you on the way. This game is a action platformer, means you jump and shoot, but to be honest, both of these things don't work in the game's favor. Except for "tiny" enemies, most of the enemies need 10 hits or more to be killed, which gets annoying very quickly, as there are also always smaller enemies like bats around you that keep you busy by flying around you in such a way so that you can't hit them while they keep hurting you.

The jumping part is also not very good, as the mechanics are rather "slippery". When you land on the ground, you often glide a few more steps, which makes precise jumps rather hard, and you will probably lose a life or two before you even get past the first screen. It's also hard to figure out, such as what is just background graphics and what is going to hurt you, as there's no obvious distinction between these things. You come in a room where three monsters walk and fly around screaming at you, and you try to avoid and shoot them, only to find out that you can't hit them. They also don't harm you, but then you go to another room and there's a lamp on a table, and just by walking over to that lamp, you get damaged and suddenly a monster comes out of the lamp, which is kind of annoying.

The graphics are ok, but not very good. The sound and music is not very interesting, and instead of cutscenes, you have voice scenes, where everything just stops, and you hear an audio recording of some kind. I cannot recommend this game.

Baby Jo - The Superhero (Going home)

Although I wanted to try each and every game for an extended period of time to see if they get better or not, I really couldn't stand this game. It was so bad that I don't even know how to describe it. The graphics are good I guess, nice and colorful, the sound and music is ok, but the gameplay is horrible!

This platformer is extremely bad in my opinion. You don't really "jump", and it feels more like you "float" when you hit the jump key. You jump really slow, like gravity does not exist, and you can change direction anyway you want and even "float" back after you jumped to the front. If you get hit by an enemy you keep losing health as long as you are "hit" by the enemy. Imagine you making a jump (float) to the right and a bee is behind you. You jump as fast as the bee flies, so you basically are inside the bee on the entire jump and keep losing health which kinda is like instant death, although it might take 5-10 seconds (did I mention you jump really slow?). Let me put it like this... stay away from this game!

Battle Field '94 in Tokyo Dome

This is Japanese wrestling game in WWF style of games. I don't like these games very much, so I can't say much about this game. The graphics are ok, but nothing special. Characters and animations are acceptable, but once again, I don't much care for these games.

Bazaru Degozaru no Game Degozaru

This is a puzzle game where you guide an ape through a level trying to catch the banana while avoiding obstacles, enemies or hazards. For this, you have different commands that you can use, jump, walk, or roll, and can arrange them in a box for later use. I don't really like this game, since the graphics are very simple and not very nice to look at. The game is in Japanese only which makes it hard to understand at times.

Browning

I wanted to like this game, but after some time with a lot of frustration with this title I decided I can't recommend it, although I played it longer than most of the other games I didn't like. I actually played 3 of the 5 stages of the game.

You play a Mecha and walk through stages, killing minor enemies or objects, and after you have killed enough of a certain type, the boss of the stage will come and attack you. The graphics, animations and sounds are all fine. There is a little bit of parallax scrolling going on in the background, the intro and

cutscenes have large animated sprites, so in all those categories, the game is nice.

Where the game lacks is in its gameplay. Your character can only shoot straight to the right the first two levels. Starting at the third level, you can turn around and also shoot to the left, but that's about it. There is no diagonal or shooting down while jumping, or even shooting up for that matter. You can jump and duck, and if you jump twice, you activate rocket boosters that allow you to fly up and down a little, but won't hold a height, so you will constantly go up and down, while heating up your rockets until they stop due to overheating (or you land and let them cool down).

Most of the time there are enemies flying around you, and since you can't shoot up, you have to jump or try to hover to hit them, which normally means you hit them once or twice and then you either to high or low again and can't hit them, but the enemies require 10 to 20 hits to kill and they respawn quickly. My guess these enemies are just there to take down your health (shield balance) and to annoy you and keep you from killing the "real enemy". They also can just crash into you for no reason except to reduce your health, even if you kill them, and when they fall towards you you get hit anyway. Touch them and you lose health. The enemy on the other hand, can fly, jump higher and farther than you can, can actually shoot diagonal, and bosses can take up to 200 hits, while one hit from them can reduce your health up to 4 points (out of a maximum of 10). So it gets frustrating very quickly, and I'm talking way too long about it.

Builder Land

In this puzzle game, you have to help a guy get to his girlfriend, who was kidnapped. For this, you have to "build the land" in front of him, meaning that you need to build bridges, and move or use objects so that he can keep going. The character is constantly walking, and the screen is constantly scrolling unless you pause it, and the only thing you can do is to pick up an item and drop it somewhere else. The graphics are rather plain, the music is very repetitive, and without pausing, the gameplay is too fast to figure everything out in time. I did not like this game.

Cyber City OEDO 808

There isn't much I can say about this game, except for it seems to be a visual novel kind of game, but since

it's completely in Japanese I can't figure out what to do and I'm not getting anywhere in the game.

ODROID-XU4 Review: Better Performance Than the Raspberry Pi, Plus USB3 and Gigabit Ethernet For USD\$60

© January 1, 2019 By Michael Larabel ODROID-XU4, Uncategorized



Hardkernel recently sent over the ODROID-XU4 for benchmarking. This ARM SBC that measures in at about 82 x 58 x 22 mm (3.23" x 2.24" x 0.87") offers much better performance than many sub-\$100 ARM SBCs, while featuring dual USB 3.0 ports, Gigabit Ethernet, eMMC storage, and compatibility with older XU3 ARM SBCs. Here's a look at the performance of the ODROID-XU4 compared to a variety of other single board computers.



Figure 1 - The ODROID-XU4 single board computer

The ODROID-XU4 is built around a Samsung Exynos5422 SoC that features four Cortex-A15 cores at 2.0GHz and four Cortex-A7 cores at 1.3GHz, while the graphics are provided by a Mali-T628. Going in hand with this Exynos 5 SoC is 2GB of LPDDR3 RAM, HDMI 1.4 supporting 1080p output, one USB 2.0 port, two USB 3.0 ports, Gigabit Ethernet, and a 30-pin

GPIO header, all contained on a PCB just 83 x 58 mm (3.23" x 2.24").

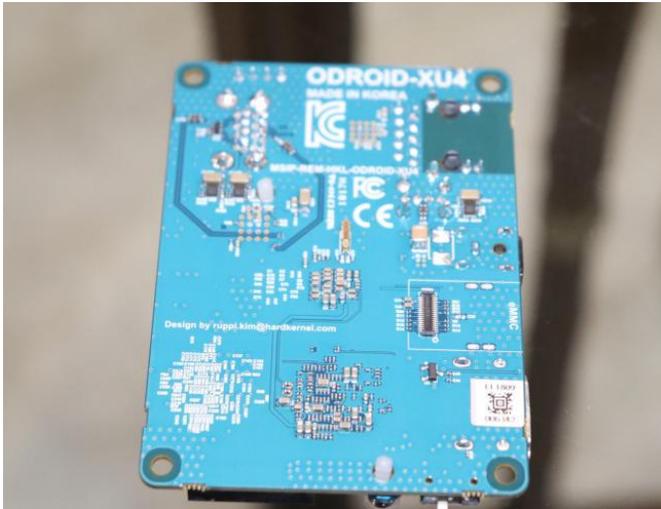


Figure 2 – The XU4 control board measures in at 83 x 58mm

(Figure 2 – The XU4 control board measures in at 83 x 58mm)

For cooling, the Exynos 5 SoC, an active heat sink, ships with this ARM SBC. There are thermal results later in this article, but the fan is certainly needed in the case of this SoC/board.

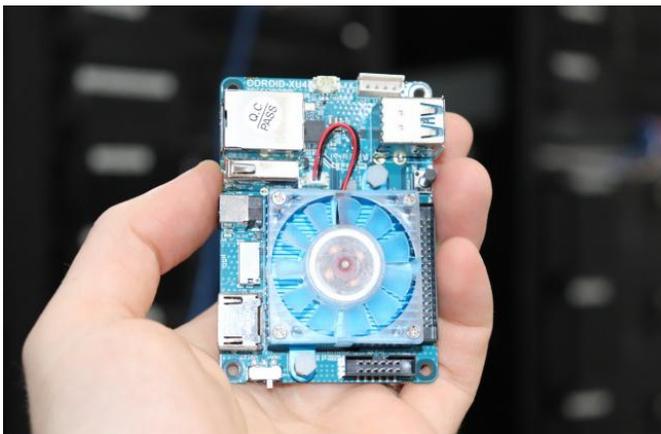


Figure 3 – The XU4 comes with a Exynos 5 SoC for cooling

The ODROID-XU4 officially supports Ubuntu 18.04 with the Linux 4.9 and 4.14 ARM kernels. There are Android images available as well as different third-party images. The ODROID Wiki continues to be quite useful for those looking for technical details and other information on Hardkernel SBCs.

My testing done was using the Ubuntu 18.04 image with the Linux 4.14.37 kernel. It uses the GCC 7.3 compiler by default. To get an idea of the performance capabilities of this ARM SBC I benchmarked it against the ASUS Tinker Board,

Raspberry Pi 3 Model A+, Raspberry Pi 3 Model B+, Jetson TX1, Jetson TX2, ODROID-C1+, ALL-H3-CC H5 2GB, and AML-S905X-CC 2GB. All of these ARM Linux benchmarks were carried out using the Phoronix Test Suite.

The XU4 memory performance came out ahead of the other low-cost ARM SBCs, but behind the NVIDIA Jetson Tegra boards that feature LPDDR4 memory. With the multi-threaded TTSIOD 3D Renderer, the ODROID-XU4 was surprisingly capable of coming out just ahead of the Jetson TX2. Just a reminder that the TX2 is a dual NVIDIA Denver core plus four ARM Cortex-A57 cores.

With the OpenMP-threaded GraphicsMagick program, the Tegra SoCs came out ahead, but the XU4 still beat out all of the other sub-\$100 ARM SBCs. The ODROID-XU4 performed very well in the 7-Zip compression test and almost matched the Jetson TX1 CPU performance.

Compile times for PHP came in between the TX1 and TX2 boards. C-Ray ray-tracing also led to the XU4 performing between the TX1 and TX2 ARM development boards.

Not being too familiar with the XU4 until receiving it, the performance potential of this ARM SBC was quite surprising. In the Redis benchmarks, the ASUS Tinker Board showed a slight advantage. For the Python benchmarking, the ODROID-XU4 came out ahead of the NVIDIA Tegra SoCs. The ODROID-XU4 also worked out well for its PHP performance compared to the other ARM hardware benchmarks.

Next up, we look at the performance-per-dollar based on the current pricing in the US for these different boards. Keep in mind that though the Jetson boards are much more expensive, they're particularly useful for GPU computing not focused on in these tests, due to predominantly testing low-end ARM SBCs in this article.

The ODROID-XU4 at \$62 offered great value and generally delivered the leading performance-per-dollar, except in some cases where the \$25 Raspberry Pi 3 Model A+ came out ahead. However, there you also have much less RAM, no USB 3.0, and no Gigabit

Ethernet. So ~\$60 is quite competitive pricing for this single board computer from Hardkernel.

As a word of caution, even with the active cooling on the ODROID-XU4, under load it can get quite hot. Here's a look at the SoC temperature with stock cooling outside of any enclosure, and then when run within the included XU4 chassis.

The average SoC temperature under load was 70C (158F) or 75C (167F) when run within the XU4 enclosure, peaking at up to 84-90C (182-194F). Between the bare setup and the enclosure, I didn't notice the enclosure causing any thermal throttling as the performance benchmark results came out identical. Warm SoCs aren't anything new but the data certainly shows active cooling is necessary with this

Exynos 5 chip. If you plan on constantly running the ODROID-XU4 under load it may be worthwhile switching to a larger heatsink/fan.

Overall, this is quite a nice ARM SBC coming in at the \$60 price point with very competitive performance. If you wish to see how your own Linux system compares to the ARM SBCs tested for this article, simply install the Phoronix Test Suite and run `phoronix-test-suite benchmark 1812082-SK-ODROIDXU403`. Thanks to Hardkernel for sending over this ODROID-XU4 for testing.

For comments, questions, and suggestions, please visit the original article at <https://www.phoronix.com/scan.php?page=article&item=ODROID-xu4-arm&num=6>.

BASH Basics: Introduction to BASH – Part 7

© January 1, 2019 By Erik Koennecke ↗ Linux



BASH

THE BOURNE-AGAIN SHELL

After all the theoretical stuff in the last few parts of this series, we're going to take a breather and deal with clever ways to increase productivity in BASH when working with ODROID. The default Ubuntu MATE system is chock-full of useful utilities without requiring so much as installing additional programs. Now, let us take this unused potential for a spin.

Remote access options

If you read this article, chances are you want to access your ODROID not just with keyboard and mouse, but with a remote console on a different system. SSH is the established standard for this– you cannot miss it since it is in every install and setup document for our single-board computers. But did you know that a full install of a graphical environment can be used even over the SSH connection?

Try to start it with:

```
ssh -X
```

when you access the system from another Linux computer. It is possible to make this work with Windows using MobaXterm, but explaining how to do this is beyond the scope of this particular article.

If you have a lightweight image viewer such as fbi, fim, or feh installed, you can now enter

```
fim /usr/share/backgrounds/ubuntu-mate-photos/
```

to browse the MATE backgrounds over the remote connection. You can also use the MATE image viewer eom if you like, but lightweight alternatives are faster over a remote connection. Advance with SPACE, quit with q.

From the command line, you can start any graphical application on your ODROID system. Since the programs are started on the ODROID, the file open dialogs also open the ODROID file system and work with the remote files.

Getting remote audio files to play locally is surprisingly difficult. Instead of using complicated setups, we cheat here by using SSHfs (install with apt install sshfs). Make or use an empty folder:

- mkdir ~/remoteODROID on your local Linux computer which you use to access the ODROID.
- With sshfs ODROID@myODROID.lan:/home/ODROID ~/remoteODROID, you can mount the ODROID home directory directly in the local ~/remoteODROID folder, without the need for setting up networking with NFS first.

Everything is mounted now and can be used like a local file so you can listen to audio files and look at video files on the ODROID without much hassle. Don't forget to unmount with

```
fusermount -u ~/remoteODROID
```

after you are done.

Running processes in the background

One peculiarity of the SSH connection is that if you close the connection, running processes get aborted. Let's say you put your new BASH scripting skills to good use and made a small script to record the temperature of the ODROID. You want to see if the system heats up too much in a new environment, stress test, or in a closed case, so you let it run for one hour:

```
templogger.sh

#!/bin/bash
# This script logs system temperature for one hour
# A file 'temp.log' in the home directory is used and overwritten

if [ -f ~/temp.log ]; then
    rm ~/temp.log      #if file exists, remove it
    touch ~/temp.log  #create file
fi
for run in {1..720} #run 720 times (5s interval)
do
    cat
    /sys/devices/virtual/thermal/thermal_zone0/temperature >> ~/temp.log
```

```
#for XU4/HC1/HC2; adjust for own system
echo $run
sleep 5
done
```

If you close the connection by shutting down the PC so you can get a cup of coffee, the test stops. To prevent this, you can background a task with `bg` and foreground it with `fg`, but backgrounding and foregrounding is cumbersome. Easier and more versatile is the program `tmux` or its predecessor `screen`. We are just touching the basics here, but you can split the screen with `tmux` and make a two- or three-pane-view, amongst other goodies. Take a look at the manual with `man tmux`, but don't get overwhelmed.

Install it as usual with `apt install tmux`. Before you start your temperature logging, start `tmux` first, just by entering `tmux`. You'll notice a status line at the bottom that contains, from left to right, the name of the current session in square brackets; the window list; the title of the active pane in double quotes; and the time and date.

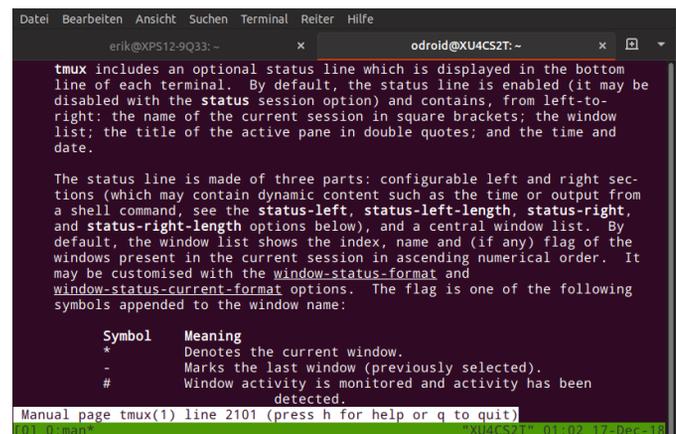


Figure 1 - A tmux screenshot

If you start the `templogger.sh` script now, you can detach the `tmux` session with `CTRL+B` and `d` for detach. Just a reminder: If you put all your scripts and programs in the `~/bin` directory, they are automatically in your path. Now close the SSH connection with `CTRL+D` and the local terminal as well with `CTRL+D`. You may even shut down your PC and get the cup of coffee I mentioned earlier.

When you remote into the ODROID again with SSH, you'll get the normal welcome screen with the message of the day. Where is your running script? A

simple tmux attach brings back the tmux session window, and you can see how many data points were already recorded in the meantime.

Repeat the procedure above to detach the session again. When you are finished, a simple quit or CTRL+D exits tmux until you need it again. Whenever you need to do operations which run for hours, or even indefinitely, tmux is a godsend. Before we explore productivity further, let's have a little fun with the visuals.

Beautification

The tmux status line is a reminder that well-placed status information is helpful for not getting lost in our system. We learned in the beginning about customizing the BASH prompt, but let's face it, all the options mentioned have been around for decades, and look accordingly. The world has evolved beyond pure ASCII now.

What can we do to have a nicely formatted console with options that look fresh and new? It's not as easy as just changing a PS3 prompt in .bashrc. However, a lot of people have already put a lot of time into this and made a nice .deb package for us to use. Install it with apt install powerline first. You'll notice that nothing appears different. We have to finish with some system modifications as well.

Open your .bashrc file with your favorite text editor and go near the end. Before adding custom functions, the standard .bashrc ends with a block starting with the following line:

```
#enable programmable completion features
```

and ending with the following line:

```
fi
```

After this block, insert the following text:

```
# Powerline features for bash, based on Ubuntu packages
if [ -f `which powerline-daemon` ]; then
    powerline-daemon -q
    POWERLINE_BASH_CONTINUATION=1
    POWERLINE_BASH_SELECT=1
.
/usr/share/powerline/bindings/bash/powerline.sh
fi
```

Save the file, exit the shell with CTRL+D and open it again. Voilà! You get a fresh look which makes working with the console and BASH even easier.

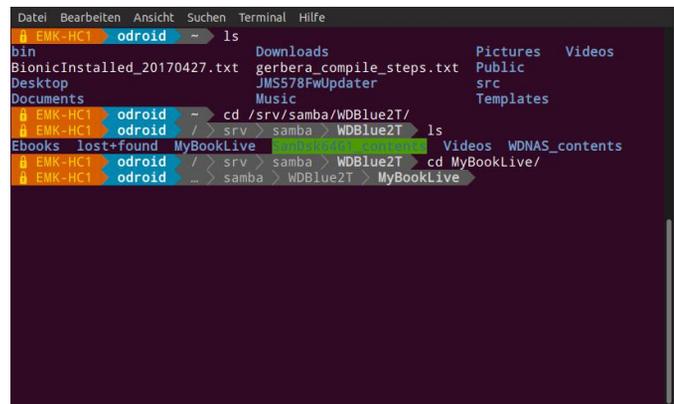


Figure 2 - Powerline BASH

References

Powerline BASH script:
<https://github.com/brujoand/sbp> Add power to your terminal with Powerline:
<https://fedoramagazine.org/add-power-terminal-powerline/>

ODROID-HC1 Swarm Cluster

© January 1, 2019 By @jit Linux, ODROID-HC1, Tutorial



The Docker team has developed a clustering and scheduling tool for Docker containers, called swarm. This article describes how one can create an ODROID-HC1 based swarm cluster. This cluster fits in a 19" rack.

3D printed 19" rack



Figure 1 - The HC1 set of devices for this article

For several months now, I have been running my applications on a home made cluster of 4 ODROID-HC1 units, running Docker containers orchestrated by swarm. The cluster is powered by a homemade power supply. I chose ODROID-HC1 and not the ODROID-MC1 because the former model offers SSD support. An SSD based system is a lot faster than one using microSD cards.

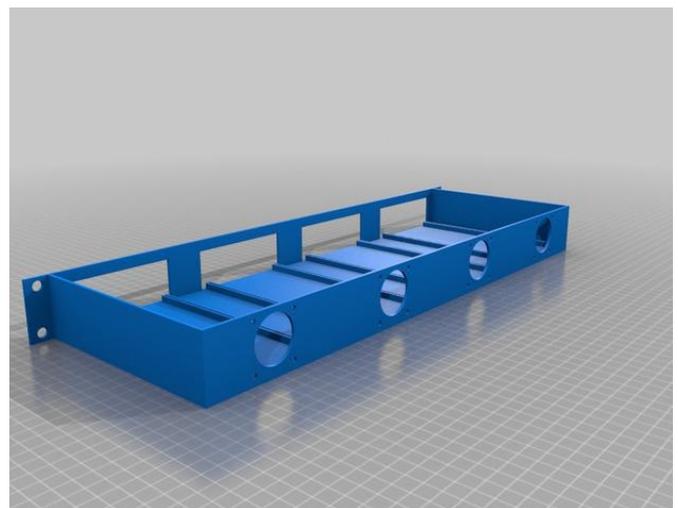


Figure 2 - 3D printed 19" rack



Figure 3 - Initial mount



Figure 4 - I soldered dupont cables to power the fan directly from the 5V input of each ODROID-HC1, as shown here.



Figure 5 - Here is how it appears with all the fans installed:



Figure 6 - The final rack setup looks like so:

Software install

I opted to install Archlinux, which can be obtained from

<https://archlinuxarm.org/platforms/armv7/samsung/odroid-xu4>. For the infrastructure management part, I opted to use saltstack. I used saltstack to "templatize" all my servers. I installed the saltstack master and minion (the NAS will be the master for all others servers), using steps I already documented at <https://www.bluemind.org/deploying-saltstack-master-minion-archlinux-arm/>

From the salt master, a simple check shows that all nodes are under control:

```
$ salt -E "node[1-4].local.lan" cmd.run 'cat /etc/hostname'
node4.local.lan:
  node4
node3.local.lan:
  node3
node2.local.lan:
  node2
node1.local.lan:
  node1
```

Rootfs on the SSD

First, partition the ssd :

```
$ salt -E "node[1-4].local.lan" cmd.run 'echo -e "
n
p
1

w
q
" | fdisk /dev/sda'
```

Format the future root partition:

```
$ salt -E "node[1-4].local.lan" cmd.run 'mkfs.ext4 -L ROOT /dev/sda1'
```

Mount ssd root partition:

```
$ salt -E "node[1-4].local.lan" cmd.run 'mount /dev/sda1 /mnt/'
```

Clone sdcard to ssd root partition:

```
$ salt -E "node[1-4].local.lan" cmd.run 'cd /;tar -c --one-file-system -f - . | (cd /mnt/; tar -xvf -)'
```

Change boot parameters so root is /dev/sda1:

```
$ salt -E "node[1-4].local.lan" cmd.run 'sed -i -e "s/root\=PARTUUID=${uuid}/root=/dev/sda1/" /boot/boot.txt'
```

Recompile boot config:

```
$ salt -E "node[1-4].local.lan" cmd.run 'pacman -S --noconfirm uboot-tools'$ salt -E "node[1-4].local.lan" cmd.run 'cd /boot; ./mkscr'
```

Reboot:

```
$ salt -E "node[1-4].local.lan" cmd.run 'reboot'
```

Remove all from sdcard, and put /boot files at root of it:

```
$ salt -E "node[1-4].local.lan" cmd.run 'mount /dev/mmcblk0p1 /mnt'$ salt -E "node[1-4].local.lan" cmd.run 'cp -R /mnt/boot/* /boot/'$ salt -E "node[1-4].local.lan" cmd.run 'rm -Rf /mnt/*'$ salt -E "node[1-4].local.lan" cmd.run 'mv /boot/* /mnt/'
```

Adapt boot.txt because the boot files are in root of boot partition and are no longer in the /boot directory:

```
$ salt -E "node[1-4].local.lan" cmd.run 'sed -i -e "s//boot///" /mnt/boot.txt'$ salt -E "node[1-4].local.lan" cmd.run 'pacman -S --noconfirm uboot-tools'$ salt -E "node[1-4].local.lan" cmd.run 'cd /mnt/; ./mkscr'$ salt -E "node[1-4].local.lan" cmd.run 'cd /; umount /mnt'$ salt -E "node[1-4].local.lan" cmd.run 'reboot'
```

Check that /dev/sda is root:

```
$ salt -E "node[1-4].local.lan" cmd.run 'df -h | grep sda'$ Node4.local.lan: /dev/sda1 118G 1.2G 117G 0% /$ Node3.local.lan: /dev/sda1 118G 1.2G 117G 0% /$ Node1.local.lan: /dev/sda1 118G 1.2G 117G 0% /$ Node2.local.lan: /dev/sda1 118G 1.2G 117G 0% /
```

SSD Benchmark

Benchmarking is complex and I am not going to say that I did it perfectly right, but at least it gives an idea of how fast an SSD can be. The SSD I used on each of the ODROID-HC1 is a 128GB Sandisk X400. I ran the following test 3 times:

```
hdparm -tT /dev/sda => 362.6 Mb/s
```

```
dd write 4k => 122 Mb/s
```

```
$ sync; dd if=/dev/zero of=/benchfile bs=4k count=1048476; sync
```

```
dd write 1m => 119 Mb/s
```

```
$ sync; dd if=/dev/zero of=/benchfile bs=1M count=4096; sync
```

```
dd read 4k => 307 Mb/s
```

```
$ echo 3 > /proc/sys/vm/drop_caches$ dd if=/benchfile of=/dev/null bs=4k count=1048476
```

```
dd read 1m => 357 Mb/s
```

```
$ echo 3 > /proc/sys/vm/drop_caches$ dd if=/benchfile of=/dev/null bs=1M count=4096
```

I tried the same test with with IRQ affinity to big cores, but it did not shown any significant impact on performance.

Finalize installation

I am not going to copy paste all my saltstack states and templates here, as it obviously depends on personal needs and tastes. Basically, my "HC1 Node" template does the following on each node:

- Change mirrorlist
- Install custom sysadminscripts
- Remove alarmuser

- Add some sysadmin tools (lsof, wget, etc.)
- Change mmc and ssd scheduler to deadline
- Add my user
- Install cron
- Configure log rotate
- Set journald config (RuntimeMaxUse=50M and Storage=volatile to lesser flash storage writes)
- Add mail ability (ssmtp)

I then changed the password for my user using saltstack:

```
$ salt "node1.local.lan" shadow.gen_password
'xxxxxx' # gives password hash in return
$ salt "node2.local.lan" shadow.set_password
myuser 'the_hash_here'
```

Finally, to ensure that no disk corruption would stop a node from booting, I forced fsck at boot time on all nodes by :

- adding "fsck.mode=force" in kernel line in /boot/boot.txt
- compile it with mkscr
- rebooting

Deploy Docker Swarm

The swarm module in my saltstack could not be recognized, despite using version 2018.3.1 modules. So, I ended up with executing the commands directly, which is not really a problem as I was not going to add a node everyday.

Build the master:

```
$ salt "node1.local.lan" cmd.run 'docker swarm
init'
```

Add worker:

```
$ salt "node4.local.lan" cmd.run 'docker swarm
join --token xxxxx
node1.local.lan:2377'
```

Add the 2nd and 3rd master for a failover ability:

```
$ salt "node1.local.lan" cmd.run 'docker swarm
join-token manager'
$ salt "node3.local.lan" cmd.run 'docker swarm
join --token xxxxx
192.168.1.1:2377'
```

```
$ salt "node2.local.lan" cmd.run 'docker swarm
join --token xxxxx
192.168.1.1:2377'
```

On checking all nodes' status with "docker node ls", we can now see one leader and 2 nodes that are "reachable". Then, I deployed a custom docker daemon configuration (daemon.json) to switch storage driver to overlay2 (the default one is to slow on xu4) and allows the usage of my custom docker registry:

```
{
  "insecure-registries":
  ["myregistry.local.lan:5000"],
  "storage-driver": "overlay2"
}
```

Docker images for the Swarm cluster

As of now, using a container orchestrator implies to either use stateless containers or to use a global storage solution. I first tried to use glusterfs on all nodes. It was working perfectly, but was way too slow (between 25 and 36 Mb/s even with optimized settings and IRQ affinity to big cores). I ended up with a simple but yet very efficient solution for my needs: An automated daily backup of all volumes on all nodes (to a network drive) An automated daily mysql database backup on all nodes (run only when mysql is detected) Containers that are able to restore their volumes from the backup during first startup An automated daily clean-up of containers and volumes on all nodes

Thus, each time a node is shut down or a stack restarted, each container is able to start on any node, retrieving its data automatically (if not stateless).

The daily backup script is shown below:

```
# monthly saved backup
firstdayofthemonth=`date '+%d'`
if [ $firstdayofthemonth == 01 ] ; then
  BACKUP_DIR="$BACKUP_DIR/monthly"
else
  firstdayoftheweek=$(date +"%u")
  if [ day == 1 ] ; then
    BACKUP_DIR="$BACKUP_DIR/weekly"
  fi
fi
```

```

volumeList=$(ls /var/lib/docker/volumes | grep
$DOCKER_VOLUME_LIST_PATTERN)

```

```

for volume in $volumeList
do
  archiveName=$(echo $volume | cut -d_ -f2-)
  mv "$BACKUP_DIR/$archiveName.tar.gz"
"$BACKUP_DIR/$archiveName.tar.gz.old"
  cd /var/lib/docker/volumes/$volume/_data/
  tar -czf $BACKUP_DIR/$archiveName.tar.gz *
2>&1
  rm "$BACKUP_DIR/$archiveName.tar.gz.old"
done

```

This is the daily cleanup script:

```

# remove unused containers and images
docker system prune -a -f

# remove unused volumes
volumeToRemove=$(docker volume ls -qf
dangling=true)

if [ ! -z "$volumeToRemove" ]; then
  docker volume rm $volumeToRemove
fi

```

Simple distributed image build

To make a simple distributed build system, I made some scripts to distribute my docker containers build across the 4 devices. All containers are then put in a local registry, tagged with the current date. Local image builder that build, tag and put in registry (script name : docker_build_image) :

```

if [ $# -lt 3 ]; then
  echo "Usage: $0 "
  echo "Example : $0 myImage armv7h
myregistry.local.lan:5000"
  echo ""
  exit 0
fi

arch="$2"
imageName="$arch/$1"
registry="$3"
tag=`date +%Y%m%d`

docker build --rm -t $registry/$imageName:$tag
-t
$registry/$imageName:latest .
docker push $registry/$imageName

```

```

docker rmi -f $registry/$imageName:$tag
docker rmi -f $registry/$imageName:latest

```

Build several images given in argument (script name:docker_build_batch) :

```

# usage : default build all
if [[ "$1" == "-h" ]]; then
  echo "Usage: $0 [image folder 1] [image
folder 2] ..."
  echo "Example :"
  echo "  build two images : $0 mariadb
mosquitto"
  echo ""
  exit 0
fi

# if any parameter, use it/them as docker
image to build
if [[ $# -gt 0 ]]; then
  DOCKER_IMAGES_DIR="@:1}"
else
  echo "Nothing to build. try -h for help"
fi

echo -e "e[1m--- going to build the following
images :"
echo -e "e[1m$DOCKER_IMAGES_DIR
"

# build and send to repository
for image in $DOCKER_IMAGES_DIR
do
  echo -e "e[1m--- start build of $image:"
  cd /home/docker/$image
  docker_build_image $image armv7h
myregistry.local.lan:5000
done

```

Distribute the builds using saltstack on the salt-master, using the previous script. The special image "archlinux" is built first if found, because all other images depend on it.

```

DOCKER_IMAGES_DIR=""
SPECIAL_NAME="archlinux_image_builder"
NODES[0]="

# usage : default build all
if [[ "$1" == "-h" ]]; then
  echo "Usage: $0 [image folder 1] [image
folder 2] ..."
  echo "Examples :"

```

```

echo " build all found images : $@"
echo " build two images : $@ mariadb
archlinux_image_builder"
echo ""
exit 0
fi

echo -e "e[1m--- Update repository (git pull)
"
# update git repository
cd /home/docker
git pull

# if any parameter, use it/them as docker
image to build
if [[ $# -gt 0 ]]; then
    DOCKER_IMAGES_DIR="${@:1}"
else
    DOCKER_IMAGES_DIR=$(ls -d */ | cut -f1 -
d'/')
fi

echo -e "e[1m--- going to build the following
images :"
echo -e "e[1m$DOCKER_IMAGES_DIR
"

# if archlinux images in array, build it first
if [[ $DOCKER_IMAGES_DIR = *"$SPECIAL_NAME"*
]]; then
    echo -e "e[1m--- found special image:
$SPECIAL_NAME, start to build
it first"
    echo -e "e[1m--- update repository on node1
"

    salt "hulk1.local.lan" cmd.run "cd
/home/docker; git pull"

    echo -e "e[1m--- build $SPECIAL_NAME image
on hulk1
"

    salt "hulk1.local.lan" cmd.run "cd
/home/docker/$SPECIAL_NAME;
./mkimage-arch.sh armv7
registry.local.lan:5000"

DOCKER_IMAGES_DIR=${DOCKER_IMAGES_DIR//$SPECIAL
L_NAME/}
fi

# update repository on all nodes

```

```

echo -e "e[1m--- update repository on node[1-
4]
"
salt -E "node[1-4].local.lan" cmd.run "cd
/home/docker; git pull"

# Prepare build processes on known swarm nodes
i=0
for image in $DOCKER_IMAGES_DIR
do
    NODES[$i]="${NODES[$i]} $image"
    i=$((i + 1))

    if [[ $i -gt 3 ]]; then
        i=0
    fi
done

echo -e "e[1m--- build plan :"
echo -e "e[1m--- node1 : ${NODES[0]}"
echo -e "e[1m--- node2 : ${NODES[1]}"
echo -e "e[1m--- node3 : ${NODES[2]}"
echo -e "e[1m--- node4 : ${NODES[3]}"
"

# distribute and launch build plan
salt "node1.local.lan" cmd.run
"docker_build_batch ${NODES[0]}"
salt "node2.local.lan" cmd.run
"docker_build_batch ${NODES[1]}"
salt "node3.local.lan" cmd.run
"docker_build_batch ${NODES[2]}"
salt "node4.local.lan" cmd.run
"docker_build_batch ${NODES[3]}"
echo -e "e[1m--- build plan finished"

```

Sources are available on GitHub

(<https://github.com/jit06/docker-images>) and Thingiverse

(<https://www.thingiverse.com/thing:3218912>). For comments, questions and suggestions, please visit the original article at

<https://www.bluemind.org/odroid-hc1-based-swarm-cluster-19-rack>.

Reference:

<https://docs.docker.com/get-started/part4/#introduction>

<https://archlinuxarm.org/platforms/armv7/samsung/odroid-xu4>

<https://wiki.archlinux.org/index.php/Saltstack>

<https://www.bluemind.org/deploying-saltstack-master-minion-archlinux-arm/>

<https://www.bluemind.org/odroid-hc1-based-swarm-cluster-19-rack>

Wireless Access Point

© January 1, 2019 By Justin Lee Linux, Tutorial



Hardkernel offers a few USB based WiFi modules to enable WiFi connectivity for their Single Board Computers (SBCs). Some of them can also be used to deploy these SBCs as Access Points. The various modules offered can be found at <https://www.hardkernel.com/product-category/connectivity/>.

Wireless Controller Information

The capability list of these modules can be obtained using the following command:

```
$ sudo iw list
...
Supported interface modes:
  * IBSS
  * managed
  * AP
  * AP/VLAN
  * WDS
  * monitor
```

* mesh point

...

If there is the "AP" capability listed in the "Supported interface modes", then that device will support the Access Point mode. Below are the details of all the modules offered by Hardkernel:

```
Wifi Module 0
Manufacturer: Mediatek Ralink
Part name: RT5370N
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 1
Band(s): 2.4GHz
Data rate: 150Mbps
MIMO configuration: 1x1:1 (1T1R)
IEEE 802.11 PHY Modes: b,g,n

Bus 005 Device 002: ID 148F:5370 Ralink
Technology, RT5370 Wireless Adapter

Wifi Module 3
```

```

Manufacturer: Realtek
Part name: RTL8188CUS
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 1
Band(s): 2.4GHz
Data rate: 150Mbps
MIMO configuration: 1x1:1 (1T1R)
IEEE 802.11 PHY Modes: b,g,n

Bus 001 Device 003: ID 0bda:8176 Realtek
Semiconductor Corp. RTL8188CUS 802.11n WLAN
Adapter

Wifi Module 4
Manufacturer: Ralink
Part name: RT5572N
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 2
Band(s): 2.4GHz, 5GHz
Data rate: 300Mbps
MIMO configuration: 2x2:2 (2T2R)
IEEE 802.11 PHY Modes: a,b,g,n

Bus 001 Device 006: ID 148f:5572 Ralink
Technology, Corp. RT5572 Wireless Adapter

Wifi Module 5
Manufacturer: Realtek
Type: chip
Number of busses: 2
Bus(es): USB 2.0 / USB 3.0
Number of bands: 2
Band(s): 2.4GHz, 5GHz
Data rate: 300Mbps
MIMO configuration: 2x2:2 (2T2R)
IEEE 802.11 PHY Modes: a,b,g,n,ac

Bus 003: ID 0bda:8812 Realtek Semiconductor
Corp. RTL8812AU 802.11a/b/g/n/ac WLAN Adapter

Wifi Module 5A
Manufacturer: Realtek
Type: chip
Number of busses: 1
Bus(es): USB 2.0
Number of bands: 2
Band(s): 2.4GHz, 5GHz
Data rate: AC600
MIMO configuration: 433 Mbps @ 5 GHz @ 1T1R /

```

```

150 Mbps @ 2.4 GHz @ 1T1R
IEEE 802.11 PHY Modes: a,b,g,n,ac

Bus 003 Device 003: ID 0bda:a811 Realtek
Semiconductor Corp.

```

Configure Access Point

- One forum member (@tam1111574) reported there is an issue with USB 3.0 port on the XU4 at <http://forum.odroid.com/viewtopic.php?f=97&t=19285>
- All commands must be executed in super user mode
- You should run `apt update && apt full-upgrade` before proceeding
- In addition, you may need to run `apt install libnl-3-dev libnl-genl-3-dev libssl-dev hostapd iptables git pkg-config vim build-essential`

We will need to configure the wireless network interface and dnsmasq daemon. The access point configuration can be divided into the following tasks:

- Setup network interface configuration
- Setup DHCP server configuration
- Setup iptables to forward the internet traffic from Ethernet to wireless lan.
- Setup hostapd server or wpa_supplicant configuration.
- Check service & WIFI configuration

Setup network interface configuration.

In order to configure the Wireless Access Point, you need to provide a static IP address for the Wireless network card. Check the wlan0 part of the following contents:

```

# vi /etc/network/interfaces
# interfaces(5) file used by ifup(8) and
ifdown(8)
# Include files from
/etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto wlan0

```

```
iface wlan0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
```

Note that the predictable interface names shown as starts with wlan if you have an interface name.

If your wireless interface name is as shown below, add net.ifnames=0 as kernel parameter to kernel command line by editing the boot.ini:

```
# ifconfig
enx7cdd9052131e Link encap:Ethernet HWaddr
7c:dd:90:52:13:1e
    UP BROADCAST MULTICAST MTU:1500
Metric:1
    RX packets:0 errors:0 dropped:321
overruns:0 frame:0
    TX packets:0 errors:0 dropped:0
overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

The local interface name issue gets resolved to have predictable name for wireless interface.

```
# ifconfig
wlan0 Link encap:Ethernet HWaddr
7c:dd:90:52:13:1e
    UP BROADCAST MULTICAST MTU:1500
Metric:1
    RX packets:0 errors:0 dropped:10
overruns:0 frame:0
    TX packets:0 errors:0 dropped:0
overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Note: The wlan0 IP address might be different from yours. Wireless interface name can be changed as it depends on the wireless cards you installed.

Setup DHCP server configuration for Access Point

Configure dnsmasq which is a lightweight DHCP and DNS server.

```
# apt install dnsmasq
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 1 reinstalled,
0 to remove and 0 not upgraded.
```

Need to get 16.2 kB of archives.

After this operation, 0 B of additional disk space will be used.

```
Get:1 http://ports.ubuntu.com/ubuntu-ports
bionic/universe armhf dnsmasq all 2.79-1 [16.2
kB]
```

Fetched 16.2 kB in 1s (16.6 kB/s)

(Reading database ... 155718 files and directories currently installed.)

```
Preparing to unpack ../dnsmasq_2.79-1_all.deb
...
```

Unpacking dnsmasq (2.79-1) over (2.79-1) ...

Setting up dnsmasq (2.79-1) ...

```
Processing triggers for systemd (237-
3ubuntu10.9) ...
```

If you faced "FAILED" message when starting up the dnsmasq.service like the below "port 53: Address already in use", then perform the following:

```
... invoke-rc.d: initscript dnsmasq, action
"start" failed...
    dnsmasq.service - dnsmasq - A lightweight
DHCP and caching DNS server
    Loaded: loaded
(/lib/systemd/system/dnsmasq.service; enabled;
vendor preset: enabled)
    Active: failed (Result: exit-code) since Mon
2018-12-10 01:59:06 UTC;
    22ms ago
    Process: 2073 ExecStart=/etc/init.d/dnsmasq
systemd-exec
    (code=exited, status=2)
    Process: 2072 ExecStartPre=/usr/sbin/dnsmasq
--test (code=exited,
status=0/SUCCESS)
    Dec 10 01:59:06 odroid systemd[1]: Starting
dnsmasq - A lightweight
DHCP and caching DNS server...
    Dec 10 01:59:06 odroid dnsmasq[2072]:
dnsmasq: syntax check OK.
    Dec 10 01:59:06 odroid dnsmasq[2073]:
dnsmasq: failed to create
listening socket for port 53: Address already
in use
    Dec 10 01:59:06 odroid dnsmasq[2073]: failed
to create listening
socket for port 53: Address already in use
    Dec 10 01:59:06 odroid dnsmasq[2073]: FAILED
to start up
    Dec 10 01:59:06 odroid systemd[1]:
dnsmasq.service: Control process
exited, code=exited status=2
```

```
Dec 10 01:59:06 odroid systemd[1]:
dnsmasq.service: Failed with
result 'exit-code'.
Dec 10 01:59:06 odroid systemd[1]: Failed to
start dnsmasq - A
lightweight DHCP and caching DNS server.
Processing triggers for systemd (237-
3ubuntu10.9) ...
```

Stop the service that was listening on port 53:

```
# Check that systemd-resolve service is
listening port 53
now(127.0.0.53:53)
# netstat -alnp | grep -w LISTEN
tcp    0    0 127.0.0.53:53    0.0.0.0:*
LISTEN 755/systemd-resolve
tcp    0    0 0.0.0.0:22      0.0.0.0:*
LISTEN 916/sshhd
tcp    0    0 127.0.0.1:631   0.0.0.0:*
LISTEN 2616/cupsd
tcp6   0    0 :::22           :::*
LISTEN 916/sshhd
tcp6   0    0 :::1:631       :::*
LISTEN 2616/cupsd

# To use the 53 port, disable & stop the
systemd-resolved service
# systemctl disable systemd-resolved.service
Removed /etc/systemd/system/multi-
user.target.wants/systemd-resolved.service.
Removed /etc/systemd/system/dbus-
org.freedesktop.resolve1.service.
# systemctl stop systemd-resolved

# dnsmasq service enable & start
# systemctl enable dnsmasq
Synchronizing state of dnsmasq.service with
SysV service script with /lib/systemd/systemd-
sysv-install.
Executing: /lib/systemd/systemd-sysv-install
enable dnsmasq

# systemctl start dnsmasq
# confirm dnsmasq service running
# netstat -alnp | grep -w LISTEN

tcp6   0    0 :::1:631       :::*
LISTEN 2616/cupsd

tcp    0    0 0.0.0.0:53     0.0.0.0:*
LISTEN 6100/dnsmasq
tcp    0    0 0.0.0.0:22     0.0.0.0:*
```

```
LISTEN 677/sshhd
tcp    0    0 127.0.0.1:631  0.0.0.0:*
LISTEN 2734/cupsd
tcp6   0    0 :::53          :::*
LISTEN 6100/dnsmasq
tcp6   0    0 :::22         :::*
LISTEN 677/sshhd
tcp6   0    0 :::1:631     :::*
LISTEN 2734/cupsd
```

Copy dnsmasq configuration file to get a backup and then make a new one.

```
# mv /etc/dnsmasq.conf /etc/dnsmasq.conf.org
# vi /etc/dnsmasq.conf
```

The dnsmasq.conf contents are like so:

```
domain-needed
bogus-priv
no-resolv
no-poll
server=/example.com/192.168.1.5
server=8.8.8.8
server=8.8.4.4
local=/example.com/
address=/doubleclick.net/127.0.0.1
no-hosts
#addn-hosts=/etc/dnsmasq.d/hosts.conf
expand-hosts
domain=example.com
dhcp-range=192.168.1.20,192.168.1.50,72h
dhcp-range=tftp,192.168.1.250,192.168.1.254
dhcp-option=option:router,192.168.1.1
dhcp-option=option:ntp-server,192.168.1.5
dhcp-option=19,0 # ip-forwarding off
dhcp-option=44,192.168.1.5 # set netbios-over-
TCP/IP aka WINS
dhcp-option=45,192.168.1.5 # netbios datagram
distribution server
dhcp-option=46,8 # netbios node type
```

Note that you can extend the dhcp-range or change the IP address in the configuration.

Setup iptables to forward the internet traffic from Ethernet to wireless LAN

Next, make port forwarding enabled automatically on boot up:

```
# vi /etc/sysctl.conf
```

Find the options below and change them:

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

Add the following contents to /etc/rc.local file in order to redirect internet traffic to wireless LAN:

```
# vi /etc/rc.local
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

You will need to update the iptables rules depending on your wireless network interface. Reboot the system to get it to work:

```
# reboot
```

Access Point Installation – hostapd

hostapd is a user space daemon for access point and authentication servers. It implements IEEE 802.11 access point management, IEEE 802.1X/WPA/WPA2/EAP Authenticators, RADIUS client, EAP server, and RADIUS authentication server. Please find more information at <https://wireless.wiki.kernel.org/en/users/documentation/hostapd>.

Below are the steps to get the hostapd compilation. Download the hostapd source code and build with proper parameters:

```
Wifi Module 0, 4, 5, 5A
# wget https://w1.fi/releases/hostapd-2.6.tar.gz
# tar xvfz hostapd-2.6.tar.gz
# cd hostapd-2.6/hostapd
/hostapd-2.6/hostapd# cp defconfig .config
/hostapd-2.6/hostapd# echo CONFIG_LIBNL32=y >> .config
/hostapd-2.6/hostapd# echo CONFIG_IEEE80211N=y >> .config
/hostapd-2.6/hostapd# echo CONFIG_IEEE80211AC=y >> .config
/hostapd-2.6/hostapd# make
```

You should download a patch file if you use Wifi Module 3 to make hostapd work for Wifi Module 3

which does not support the standard nl80211 driver from hostapd on kernel 3.10, 3.14, 3.16.

```
Wifi Module 3
# git clone https://github.com/pritambaral/hostapd-rtl871xdrv.git
# wget https://w1.fi/releases/hostapd-2.6.tar.gz
# tar xvfz hostapd-2.6.tar.gz
# cd hostapd-2.6
/hostapd-2.6# patch -p1 < ../hostapd-rtl871xdrv/rtlxdrv.patch /hostapd-2.6# cd hostapd /hostapd-2.6/hostapd# cp defconfig .config /hostapd-2.6/hostapd# echo CONFIG_LIBNL32=y >> .config
/hostapd-2.6/hostapd# echo CONFIG_DRIVER_RTW=y >> .config
/hostapd-2.6/hostapd# make
```

Backup the current hostapd and replace the file with the configured one:

```
/hostapd-2.6/hostapd# mv /usr/sbin/hostapd /usr/sbin/hostapd.org
/hostapd-2.6/hostapd# cp hostapd /usr/sbin/
```

Verify that you have installed the latest version:

```
/hostapd-2.6/hostapd# cd ~
# hostapd

hostapd v2.6
User space daemon for IEEE 802.11 AP management,
IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator
Copyright (c) 2002-2016, Jouni Malinen <j@w1.fi> and contributors
```

Configure hostapd for Access Point use:

```
# vi /etc/default/hostapd
```

Find the options below and edit:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
DAEMON_OPTS="-B"
```

You can update this DAEMON_OPTS to get the logs from hostapd daemon by editing the /etc/hostapd/hostapd.conf file:

```
# vi /etc/hostapd/hostapd.conf
```

The contents of the hostapd.conf files for 2.4 GHz configuration is as follows:

```
# HostAPD <2.4 Ghz configuration hostapd.conf
file>
# Interface
interface=wlan0
# driver
driver=nl80211
# Logging
logger_syslog=-1
logger_syslog_level=3
logger_stdout=-1
logger_stdout_level=2
# CTRL-Interface
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
# WLAN
country_code=KR
ssid=OdroidAPn
hw_mode=g
channel=6
beacon_int=100
dtim_period=2
max_num_sta=255
rts_threshold=2347
fragm_threshold=2346
preamble=1
# WPA2
wpa=2 # WPA2 only
wpa_passphrase=hardkernel
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
auth_algs=3 # 1=wpa, 2=wep,
3=both
macaddr_acl=0
wmm_enabled=1
eap_reauth_period=360000
fragm_threshold=2346
rsn_preauth=1
rsn_preauth_interfaces=wlan0
wpa_group_rekey=600
wpa_ptk_rekey=600
wpa_gmk_rekey=86400
# N-WLAN
ieee80211n=1
ht_capab=[HT20+][SHORT-GI-20][DSSS_CCK-20]
require_ht=0
obss_interval=0
```

The contents of the hostapd.conf files for 2.4GHz configuration for WiFi Module 3 is as follows:

```
# interface
interface=wlan0
# driver
driver=rtl871xdrv
# CTRL-Interface
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
# SSID
ssid=OdroidAP
# WLAN
country_code=KR
hw_mode=g
channel=1
beacon_int=100
dtim_period=2
max_num_sta=255
rts_threshold=2347
fragm_threshold=2346
preamble=1
# Logging
logger_syslog=-1
logger_syslog_level=3
logger_stdout=-1
logger_stdout_level=2
# passphrase
wpa_passphrase=hardkernel
# WPA2
wpa=3
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
beacon_int=100
auth_algs=3 # 1=wpa, 2=wep,
3=both
macaddr_acl=0
wmm_enabled=1
eap_reauth_period=360000000
rsn_preauth=1
rsn_preauth_interfaces=wlan0
wpa_group_rekey=600
wpa_ptk_rekey=600
wpa_gmk_rekey=86400
```

The contents of the hostapd.conf files for 5 GHz Realtek RTL8812AU chipset is as follows:

```
# HostAPD <5.0 Ghz configuration hostapd.conf
file>
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
```

```

interface=wlan0
driver=nl80211
### IEEE 802.11
ssid=Odroid5Ghz
hw_mode=a
channel=36
max_num_sta=128
auth_algs=1
### DFS
country_code=KR
ieee80211d=1
ieee80211h=1
#ieee80211ac=1
### IEEE 802.11n
ieee80211n=1
ht_capab=[HT40+][SHORT-GI-20][SHORT-GI-40]
[DSSS_CCK-20]
### IEEE 802.11ac
#ieee80211ac=1
#vht_oper_chwidth=1
#vht_capab=[SHORT-GI-40][RXLDPC][TX-STBC-2BY1]
#vht_oper_centrfreq_seg0_idx=0
### IEEE 802.11i
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=hardkernel
rsn_pairwise=CCMP
### hostapd event logger
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=2
### WMM
wmm_enabled=1
uapsd_advertisement_enabled=1
wmm_ac_bk_cwmin=4
wmm_ac_bk_cwmax=10
wmm_ac_bk_aifs=7
wmm_ac_bk_txop_limit=0
wmm_ac_bk_acm=0
wmm_ac_be_aifs=3
wmm_ac_be_cwmin=4
wmm_ac_be_cwmax=10
wmm_ac_be_txop_limit=0
wmm_ac_be_acm=0
wmm_ac_vi_aifs=2
wmm_ac_vi_cwmin=3
wmm_ac_vi_cwmax=4
wmm_ac_vi_txop_limit=94
wmm_ac_vi_acm=0
wmm_ac_vo_aifs=2
wmm_ac_vo_cwmin=2
wmm_ac_vo_cwmax=3

```

```

wmm_ac_vo_txop_limit=47
wmm_ac_vo_acm=0
### TX queue parameters
tx_queue_data3_aifs=7
tx_queue_data3_cwmin=15
tx_queue_data3_cwmax=1023
tx_queue_data3_burst=0
tx_queue_data2_aifs=3
tx_queue_data2_cwmin=15
tx_queue_data2_cwmax=63
tx_queue_data2_burst=0
tx_queue_data1_aifs=1
tx_queue_data1_cwmin=7
tx_queue_data1_cwmax=15
tx_queue_data1_burst=3.0
tx_queue_data0_aifs=1
tx_queue_data0_cwmin=3
tx_queue_data0_cwmax=7
tx_queue_data0_burst=1.5

```

On the Wifi module 4 and Wifi module 5, Wifi module 5A, we need to tune these setting depending on the ht_capab. Replace the country_code, ssid, and wpa_passphrase as per your requirement. N-WLAN is optional and can be left out.

Check your hostpad & dnsmasq service status

The WiFi IP address must be the same of the set for /etc/network/interfaces. You have to allocate specific IP address you set into the /etc/network/interfaces file as a static method to wlan0 interface.

```

# Force allocation of IP address
# ifconfig wlan0 192.168.1.1

# Check IP address
# ifconfig wlan0
wlan0:
flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
mtu 1500
        inet 192.168.1.1 netmask 255.255.255.0
broadcast 192.168.1.255
        ether 40:a5:ef:f3:98:6a txqueuelen 1000
(Ethernet)
        RX packets 122 bytes 13344 (13.3 KB)
        RX errors 0 dropped 1 overruns 0
frame 0
        TX packets 49 bytes 18722 (18.7 KB)
        TX errors 0 dropped 473 overruns 0
carrier 0 collisions 0

```

After you finish all of the setup steps, restart the services and check their statuses. Enter the following code to enable by default:

```
# update-rc.d hostapd enable
```

Restart the services:

```
# service hostapd restart
# service dnsmasq restart
```

If you get this message when starting hostapd service, "Failed to start hostapd.service: Unit hostapd.service is masked.", unmask that service and try it again:

```
systemctl unmask hostapd.service
```

Check the services:

```
# service hostapd status
# service dnsmasq status
```

Check the active (running) status:

```
# hostapd
hostapd.service - LSB: Advanced IEEE 802.11
management daemon
   Loaded: loaded (/etc/init.d/hostapd; bad;
vendor preset: enabled)
   Active: active (running) since Sun 2017-07-
09 19:48:41 UTC; 31s ago
   Docs: man:systemd-sysv-generator(8)
   Process: 789 ExecStart=/etc/init.d/hostapd
start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/hostapd.service
           └─798 /usr/sbin/hostapd -B -P
/run/hostapd.pid -B /etc/hostapd/hostapd-
2.6.conf

Jul 09 19:48:41 odroid systemd[1]: Starting
LSB: Advanced IEEE 802.11 management daemon...
Jul 09 19:48:41 odroid hostapd[789]: *
Starting advanced IEEE 802.11 management
hostapd
Jul 09 19:48:41 odroid hostapd[789]:
...done.
Jul 09 19:48:41 odroid systemd[1]: Started
LSB: Advanced IEEE 802.11 management daemon.

# dnsmasq
dnsmasq.service - dnsmasq - A lightweight DHCP
and caching DNS server
   Loaded: loaded
(/lib/systemd/system/dnsmasq.service; enabled;
```

```
vendor preset: enabled)
   Drop-In:
/run/systemd/generator/dnsmasq.service.d
           └─50-dnsmasq-$named.conf, 50-
insserv.conf-$named.conf
   Active: active (running) since Sun 2017-07-
09 19:48:31 UTC; 4min 36s ago
   Process: 592
ExecStartPost=/etc/init.d/dnsmasq systemd-
start-resolvconf (code=exited,
status=0/SUCCESS)
   Process: 575 ExecStart=/etc/init.d/dnsmasq
systemd-exec (code=exited, status=0/SUCCESS)
   Process: 554 ExecStartPre=/usr/sbin/dnsmasq
--test (code=exited, status=0/SUCCESS)
   Main PID: 591 (dnsmasq)
   CGroup: /system.slice/dnsmasq.service
           └─591 /usr/sbin/dnsmasq -x
/var/run/dnsmasq/dnsmasq.pid -u dnsmasq -r
/var/run/dnsmasq/resolv.conf -7
/etc/dnsmasq.d, .dpkg-dist, .dpkg-old,

Jul 09 19:48:30 odroid dnsmasq[591]: DNS
service limited to local subnets
Jul 09 19:48:30 odroid dnsmasq[591]: warning:
ignoring resolv-file flag because no-resolv is
set
Jul 09 19:48:30 odroid dnsmasq-dhcp[591]:
DHCP, IP range 192.168.1.250 -- 192.168.1.254,
lease time 1h
Jul 09 19:48:30 odroid dnsmasq-dhcp[591]:
DHCP, IP range 192.168.1.20 -- 192.168.1.50,
lease time 3d
Jul 09 19:48:30 odroid dnsmasq[591]: using
local addresses only for domain example.com
Jul 09 19:48:30 odroid dnsmasq[591]: using
nameserver 4.4.4.4#53
Jul 09 19:48:30 odroid dnsmasq[591]: using
nameserver 8.8.8.8#53
Jul 09 19:48:30 odroid dnsmasq[591]: using
nameserver 192.168.1.5#53 for domain
example.com
Jul 09 19:48:30 odroid dnsmasq[591]: cleared
cache
Jul 09 19:48:31 odroid systemd[1]: Started
dnsmasq - A lightweight DHCP and caching DNS
server.
lines 1-22/22 (END)
```

Access Point Installation – wpa_supplicant

The wpa_supplicant application is a user space application which works as a WPA supplicant and SME

(to handle initiating MLME commands). Please refer to the link for further informations: https://wireless.wiki.kernel.org/en/users/documentation/wpa_supplicant.

Below are the steps to get the wpa_supplicant compilation. First, download the wpa_supplicant source code and build it:

```
# wget https://w1.fi/releases/wpa_supplicant-2.6.tar.gz
# tar xvfz wpa_supplicant-2.6.tar.gz
# cd wpa_supplicant-2.6/wpa_supplicant/
/wpa_supplicant-2.6/wpa_supplicant# cp defconfig .config

# Enable AP, P2P and WPS needed to configure wpa_supplicant in AP mode.
/wpa_supplicant-2.6/wpa_supplicant# echo CONFIG_P2P=y >> .config
/wpa_supplicant-2.6/wpa_supplicant# echo CONFIG_AP=y >> .config
/wpa_supplicant-2.6/wpa_supplicant# echo CONFIG_WPS=y >> .config
/wpa_supplicant-2.6/wpa_supplicant# echo CFLAGS +=-I/usr/include/libnl3/ >> .config
/wpa_supplicant-2.6/wpa_supplicant# echo CONFIG_LIBNL32=y >> .config
/wpa_supplicant-2.6/wpa_supplicant# make
```

Backup the current wpa_supplicant and replace the file with configured one:

```
/wpa_supplicant-2.6/wpa_supplicant# mv /sbin/wpa_supplicant /sbin/wpa_supplicant.org
/wpa_supplicant-2.6/wpa_supplicant# cp ./wpa_supplicant /sbin/
```

Verify that you have installed the latest version:

```
/wpa_supplicant-2.6/wpa_supplicant# cd ~
# wpa_supplicant -v

wpa_supplicant v2.6
Copyright (c) 2003-2016, Jouni Malinen
<j@w1.fi> and contributors
```

Create a configuration file for wpa_supplicant to run in AP mode call wpa.conf:

```
# vi wpa.conf
```

The wpa.conf file contents is as follows:

```
network={
    ssid="odroid-WPA"
    mode=2
    key_mgmt=WPA-PSK
    psk="hardkernel"
    frequency=2437
}
```

Reboot the system.

```
# reboot
```

Note that you can change the frequency range between 2.4 GHz and 5 GHz.

```
# ifconfig wlan0
wlan0:
flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
mtu 1500
    inet 192.168.1.1 netmask 255.255.255.0
    broadcast 192.168.1.255
    ether 40:a5:ef:f3:98:6a txqueuelen 1000
(Ethernet)
    RX packets 122 bytes 13344 (13.3 KB)
    RX errors 0 dropped 1 overruns 0
    frame 0
    TX packets 49 bytes 18722 (18.7 KB)
    TX errors 0 dropped 16 overruns 0
    carrier 0 collisions 0
```

This is an example of starting wpa_supplicant as Access Point:

```
# wpa_supplicant -Dnl80211 -iwlan0 -c ~/wpa.conf
Successfully initialized wpa_supplicant
Using interface wlan0 with hwaddr 7c:dd:90:52:13:1e and ssid "odroid-WPA"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: CTRL-EVENT-CONNECTED - Connection to 7c:dd:90:52:13:1e completed [id=0 id_str=]
```

Start Wpa_supplicant as a server

Append the following in the /etc/rc.local file:

```
sudo service network-manager stop
wpa_supplicant -B -Dnl80211 -iwlan0 -c /root/wap.conf
```

For comments, questions and suggestions, please visit the original article at

https://wiki.odroid.com/accessory/connectivity/wifi/wireless_ap_mode.

Meet An ODROIDian: Justin Lee, CEO of Hardkernel

© January 1, 2019 By Justin Lee Meet an ODROIDian



How has Hardkernel grown over the past 5 years? It is really hard to remember what has happened in the past 5 years due to the limitations of my brain capacity, but let me try to recall what changes have happened. 5 years ago, Hardkernel had 11 people including me, but we have 26 members today. We go to workshops two to three times a year to relax and recharge.



Figure 1 - Hardkernel at a workshop 5 years ago



Figure 2 - Hardkernel at a recent workshop

We shipped around 3,000 ODROID SBC products per month in 2014. We've been shipping over 15,000 boards per month in 2018. Our global community forum had slightly over 4,000 registered members in 2014 and there are over 26,000 members today.

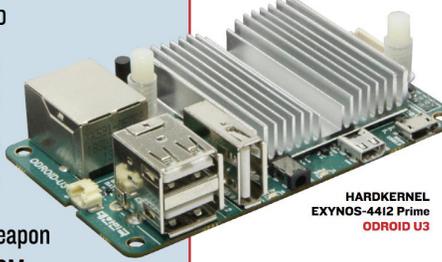
We started publishing the ODROID Magazine in January 2014. This month is the first issue of the 6th year amazingly. I didn't know how much we had grown over the past 5 years!

ODROID Magazine

If you like the ODROID-U2, you'll love our new ODROID

U3

hardkernel's new weapon for developing on ARM
And it is amazing!



HARDKERNEL EXYNOS-4412 Prime ODROID U3

RPi vs U3
We show you how they stand against each other

Ubuntu 13.10

Install and troubleshooting (the easy way)

- Use C/C++/Python/Java and IDE to: Hack your hardware and control LEDs
- Start programming right away!
- High Performance Computing with multiple ODROIDS

- Meet an Odroidian Justin Lee, CEO of Hardkernel
- Linux Gaming
- Real-Time Mobile Eye Tracking
- Introduction to the HTML5 Video Player

Figure 3 – ODROID Magazine’s first issue cover

What new directions do you see Hardkernel moving in the next 5 years? There is nothing new. We just want to continue to create products that will give developers a lot of fun, so we will keep closely communicating with our community members as we have always done.



Figure 4 – Hardkernel listens to its users to constantly improve its products

How has the ODROID community contributed to Hardkernel's success? The community has always given us lots of great ideas to create new and cool products. Forum members kindly help us to resolve many software and hardware issues as well. We still have very limited internal human resources, but we've overcome it thanks to our great community.

What steps does the Hardkernel team go through to develop new products? There is nothing special. If we find a bright idea with good feasibility, we just build a prototype quickly to test the functionality, performance and usability. If the test results are acceptable, we polish the prototype to make it more attractive to developers. After a couple of iterations of prototyping, we build engineering samples to validate our manufacturing process. We also give away some engineering samples to important community members to garner their feedback in many cases. If there are no critical issues, we just launch the new product finally. But there are a lot of hidden time consuming internal processes and mistakes because we are human beings.

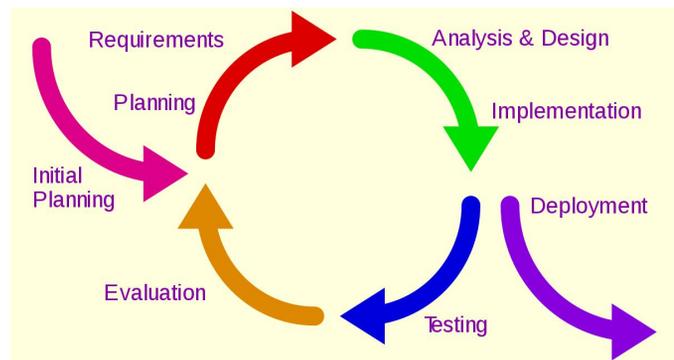


Figure 5 – Hardkernel’s product development cycle

ODROIDS have become synonymous with retro gaming innovation; was that intended? ODROID Game Station Turbo (OGST), ODROID Retro Arena (ORA), Batocera, Recalbox, Lakka, and other gaming-oriented, community-driven OS images for the ODROID-XU4 have improved significantly and been very popular. Recently, the ODROID-XU4 SBC price dropped again down to \$49, making it the most affordable and powerful platform for building a retro gaming console. Frankly speaking, we always like to play various games too, and it is one of the main reasons why we made a Nintendo64-like game console kit as well as the amazing portable gaming device called the ODROID-GO!



Figure 6 - Nintendo 64 emulation is very popular with ODROIDians

What personal projects do you use ODROIDS for? I've been using several ODROID boards on my desk for many different purposes. Here is a good example: I've used an Intel Xeon E3-1230-based desktop PC since late 2013, and the hardware system has had some stability issues lately. Fortunately, our brand new ODROID-H2 performance is acceptable for my tasks, and I decided to change my computer. Let me compare the old and new systems in Table 1.

| | Old PC | | New PC | |
|--------------|-------------------------------------|--------------|---------------------------------------|--------------|
| | Specification | Price(2013) | Specification | Price(2018) |
| CPU | Xeon E3-1230 v3 Quad-core 3.3Ghz | \$420 | Gemini Lake J4105 Quad-core 2.4Ghz | \$111 |
| Motherboard | Asrock B75M-DGS | | Hardkernel ODROID-H2 | |
| Graphics | MSI Radeon 7750 1G | | On-chip Intel UHD 600 | |
| RAM | 2 x 4G DDR3-1600 | \$50 | 2 x 4G DDR4-2400 | \$72 |
| Storage | Plextor SATA SSD 256G | \$300 | Kingston NVMe SSD 240G | \$57 |
| Power Supply | PC-ATX 400W | \$25 | DC Adapter 60W | \$11 |
| Case | Mid-Tower | \$25 | Type-2 Kit | \$7 |
| Total | | \$820 | | \$259 |

Table 1 - Justin's old PC and new ODROID-H2 component and price comparison

The price comparison is not fair since the SSD price has dropped so much, but we have to agree that the new ODROID-H2 PC is still very affordable, with no noise from cooling fan, and great energy savings when compared to my old Xeon PC. The Xeon E3 processor computing power is obviously 40~50% faster than J4105 SoC, but the much faster DDR4 RAM and NVMe flash storage in my ODROID-H2 system improves the overall performance.

I use two 2560×1440 27" 4K monitors for my tasks. I ran Firefox on the left side and Chrome on the other side. Even after I opened 25 tabs on each browser, the user experience is still very smooth and comfortable.

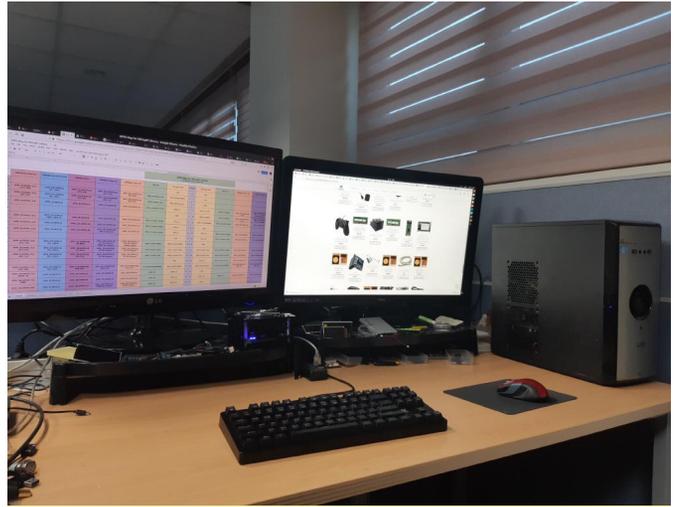


Figure 7 - Justin's desk before introducing the ODROID-H2

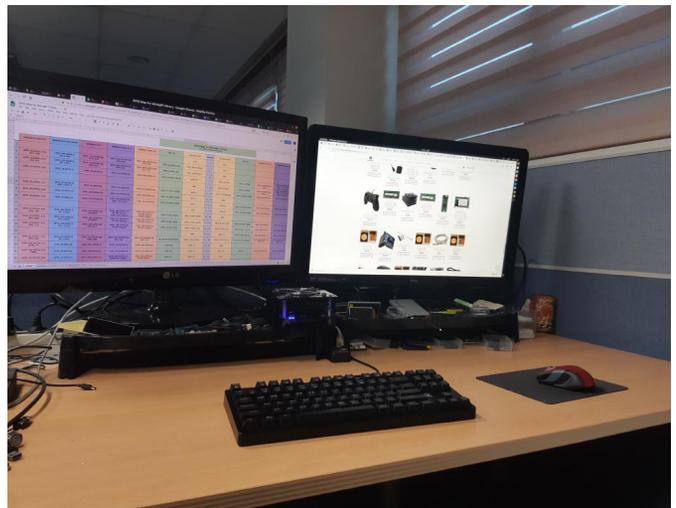


Figure 8 - Justin's desk after adding the ODROID-H2

My old and bulky computer occupied a lot of space on my desk. It may be difficult to spot my new mini PC. I can also place more ODROIDS on my desk now. Can you find my slightly customized H2 case in the picture?



Figure 9 - A closeup of Justin's customized ODROID-H2 case

