



1/edit/one.o0/the_first_issue

the first issue, ed has been wanting to do this since they were ~15 reading gaming mags every month and browsing the newagents shelves until they were kicked out for not buying anything.

running to the bins at the back of the shopping center to find the coverless mags that were thrown out because they didn't sell on a thursday arvo.

currently thinking this will be a monthly edition zine. it'll be pretty loose but you should be safe to expect a release by the 25th of each month mistakes and all.

in this first issue we look at a number of cool things we've played with of late.

chaiki for the ps4, rpi as a desktop, ed picked up an oscc, compiling xfce.

ed is playing pokemon blue on the gb through retroarch so expect a short "8hrs in (now 20hrs)" review.

if you expect in depth stuff you are in the wrong place, hell look at the rpi on the cover, electrical tape and all ... this is just a kick back and flick through style zine.

its not intended to be a burden to write for nor manage. all in all though its supposed to be an informative fun monthly.

hell i doubt even the layout will be anything similar from month to month until we find our stride

we've given it a crack anyway, see how it goes

- ed

this issue

1/edit

the first issue, and rambling from ed.

2/re/

A review of Pokemon Blue for the gb.

3/poc/

chaiki an opensource alternative for ps4 remote play.

4/laz/

we look at the oscc, this thing is incredible.

5/boot/

ed busted the i7 2600k and used an rpi as a desktop throughout dec 2020, this is that story.

6/doc/

ansible ... just ansible, it's not puppet, taking a quick look at it and how it works.

cover: rpi desktop
author: sairuk
date: 2019-10-13

2/re/1998/gb/a_look_back_at_pokemon_blue

never played this when i was younger so now that we're all locked up in iso seemed like as good of a time as any to give it a go.

i have next to no experience with the pokemon series of games so we can set that as a baseline.

the game starts off with a visit to professor oak who gives you a basic pokemon out of a selection of 3 available in his lab. your 'rival' red, who is also professor oak's grandson rags on you without any word of discipline.

with pokedex and pokemon in hand you head off from your home town to find and capture any number of innocent pokemon. once caught a pokedex entry will be created for the pokemon. the pokedex will help to track the pokemon you meet and have. it will also be graded by oak throughout the game.



pokemon have multiple levels, abilities and evolutions. there are a tonne of



environments to explore, pokemen to capture with no shortage of items and strategies to take into battle.

make no mistake this game is addictive, i've already sunk 8hrs into this without batting an eyelid, and it hits every note once you get into it.

there are enough rpg style elements there to disguise what is effectively a gambling game and it's fun. almost tragically i hate to admit it but it is fun.

i can't see this making my top 10 any time soon but its something i don't mind picking up and advancing on a casual basis. for me having this running on my phone through retroarch has turned into a welcome distraction from the relentless low quality offerings on the play store.

- sairuk



3/poc/now/ps4/chiaki_an_opensource_client_for_ps4_remote_play

the ps4(5) offers remote play on pc and phone(s) which is something i find myself using quite a lot when the front tv is busy with other things.

linux, android, macos and nintendo switch. I also build this successfully on raspbian (rpi4).

sony produce a pc client that you can use to connect to the ps4, if you are not in the general recognised 'pc demographic'(linux) you cannot run this application because unfortunately it will not successfully run on wine.

the setup is a little iffy needing to run a python script to obtain your psn id that can



the open source community provides (with varying success sometimes) where necessary and to that end we are given chiaki and open source alternative ps4 remote play client unofficial but better in every respect.

be inserted into the application but as of writing the code for it seemed harmless and once done you can store the info and never need to run it again.

chaiki handles multiple devices and official controllers putting it in league with the official offering but its what chaiki does beyond this where things get interesting.

this software does one job very well and should be on your list of software to have a look at.

- sairuk

chaiki allows for the multiple unofficial controllers to be used, it supports multiple resolutions and keyboard mapping as a controller and this alone puts it well beyond the feature set of the official offering.

want to write something?

contact us with your idea.
 : tech, gaming, retro stuff
 : anything else (discuss)
 : we are after cover art.
 : submissions welcomed

@@scrapezine_at_mameau.com@@

chiaki is crossplatform with builds available for windows,



4/laz/now/rgb/open_source_scan_convertor

putting an older console on a newer display panel is a variable experience at best. on new digital panels, crts on the other hand did what they wanted. the end result may at best be 'usable'.

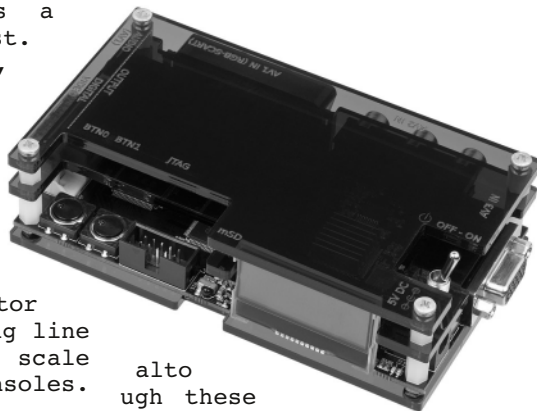
In search of a real solution markus hienkari gave us the open source scan convertor (oss) which is a zero lag line doubler device that can scale up to 5x for most consoles.

inputs are available for vga, scart and component and revision 1.6 of the oss outputs to hdmi, prior versions output to dvi. these inputs should cover all your needs.

the oss doesn't do any signal management at all, keep in mind games like resident evil or chrono cross used different video modes between ingame and menu. when dealing with the game resolutions the display may switch resolution. thankfully the affected list of games is small and solutions (gbs8200) do exist to correct this

the device can manipulate the output in to many ways to cover here you can manipulate the generic line doubled content per console to achive a near crt perfect look and then (as of firmware 1.8) store up to 15 profiles on the device to recall later or leave it as is.

community member firebranx provides a web editor and exporter for profiles and a precanned 'optimal pack',



alto though these seem targeted to target ntsc consoles currently and didn't work well with pal.

an important point here is the ossc design is opensource so you can grab the bom and build it yourself, if not cheap preassembled knockoffs exist but be mindful of the parts going into it might not be the highest quality so an official reseller is probably the best bet.

this is really cool device and the experience is chalk and cheese when you run without it, i can't say why it took me so long to buy one. i guess without the experience the price of a preassembled one hurt at \$230 it was no small gamble.

markus has since announced the fpga based ossc pro which we are keen to check out.
- sairuk

5/boot/now/rpi/rpi4_as_a_desktop

i bent some cpu pins while cleaning of my old mobo, two broke when trying to fix them, game over man.

after 8yrs of reliable service my i7 was dead int he water. this of course happened over christmas when no-one was shipping anything. i had an rpi4 spare, so time to test the desktop experience.

this was a fun experience, not going to lie. it was like having a computer for the first time and just flat out exploring what was possible.

hardware, i run a UASP usb drive with powered hdmi to vga adapters and an externally pow

ere
d
usb
hub
.

li
nux
,

the main os for the rpi was already my daily driver so wasn't a big leap, had i been a windows user this new eco would have been challenging.

the goals i set out were to run dual head, ssh/rdp, web stuff, devel scripts and of course gaming/emulation.

the ssh/rdp work was a no-brainer the rpi can handle this workload easily

there was some work getting the system up with vulkan support, compiling mesa and moving to xfce (my chosen de) at a later version etc.



gaming, vkquake3, box86, retroarch, ppsspp, chaiki, steamlink are all available for the

rpi plus more. box86 opened the door to a larger catalog than native ports. there are some caveats but all in all it becomes a viable gaming platform.

there are some resourcing issues on the web front, the older firefox-esr is useless, chromium is shipped. most sites these days are so bloated chromium maxes out all 4 cpus in general usage. hardware acceleration helps slightly but its a ways away from desktop performance.

overall though the rpi is close, so close in fact that i considered having it for my daily driver permanently.

read the full experience on mameau.com

- sairuk



```
File Edit View Terminal Tabs Help
1 [|||||]100.0%
2 [|||||]100.0%
3 [|||||]98.8%
4 [|||||]100.0%
s: Mem[|||||] 939M/3.56G
t: Swp[|||||] 16.5M/100.0M
d:
```

6/doc/now/ansible

ansible, what a great little (big) tool for almost everything.

as a brief really introduction to ansible we will touch on some key concepts. this should be enough to get you started.

ansible is a python based configuration management tool that takes in a playbook (generally) and runs it concurrently across as many hosts as you define.

A playbook is a set of defined plays in yaml format which translate to commands to be run on a target system through ansible modules.

When writing plays you will leverage a number of pre-existing ansible modules. if you can't find what you need you can lean on a generic module to do what you need, or write your own.

An inventory is a setup fo deifined hosts or groups of hosts or servers which can be targeted in a playbook.

Once defined if you need to make a change to all these hosts you can update your ansible playbook and run it against your inventory to ensure configuration consistency.

Automating the deployments of something out of left field, say retroPie can be an interesting activity. So lets install some binary packages with ansible

Lets take a look at that

Create the yaml file header

```
---
- name: RetroPie packages
  hosts: all
  gather_facts: false
```

Create a list var

```
vars:
  retroPie_packages:
    - "emulationstation-dev"
```

Create a task

```
task:
- name: "Install binary packages"
  command: "/home/pi/RetroPie-Setup/retroPie_packages.sh {{ item }}"
  install_bin
  become: true
```

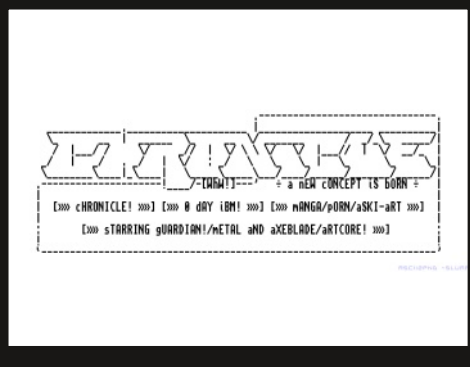
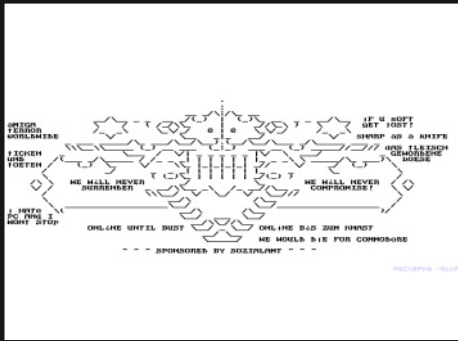
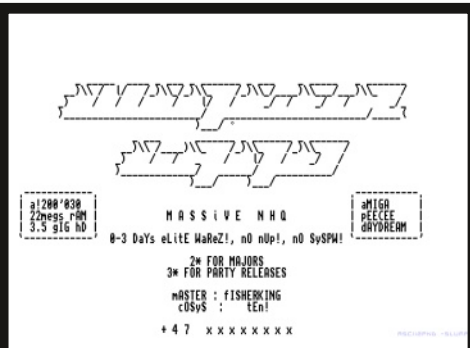
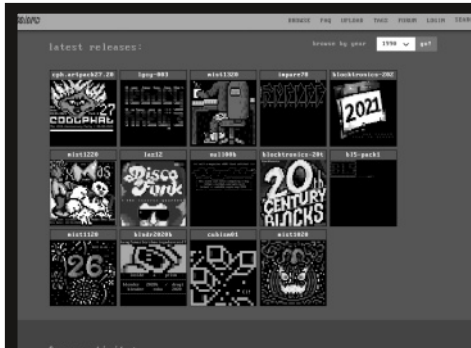
This will install the binary package for emulationstation-dev everytime the playbook is run and any other packages you were to add to the retroPie_packages list

You can now run this file against your retroPie setups with

```
$ ansible-playbook <yaml file>
```

This is a very slimed down example of what you could use ansible for but is something I have setup alongside nfs rom mounts and a host of other customizations for retroPie.

There are other options out there for configuration management but from what I've played with ansible is king. It requires python on the target host and nothing else.
- sairuk



About

Members

ed - sairuk (@sairukau)

Greets (no order)

slateman
 blahjedi
 blocktronics
 bob
 you

Creds

blocktronics

- <http://blocktronics.org/>

chiaki

- <https://git.sr.ht/~thestr4ng3r/chiaki>

retrorgb

- <https://www.retrorgb.com>

mameau

- <https://www.mameau.com/linux/rpi-desktop>

ansible

- <https://www.ansible.com/>