

Batocera native CRT output



This guide only applies to Batocera v32. For v30/v31, refer to the [legacy guide](#).

Batocera has the capability to natively output an analogue signal, should you have the hardware to do so. The specifics are later in this guide, but that basically means you have a VGA/DVI-I port on your graphics card (which is becoming exceedingly rare with modern graphics cards). CRT displays generally accepted a 15.7-16.5 kHz signal (this is not the screen refresh rate, this is the individual bits of data transferred per second), while modern digital standards use higher rate signals at 70 kHz and above. Modern cards with only digital outputs have a minimum signal floor starting at 70 kHz, and thus won't be compatible for use in this guide.



You can however use an active adapter to convert a digital signal to an analogue one (this will not be native), but that is outside the scope of this guide and you won't get support for such external hardware using these configurations.

Recommended reading:

- [RetroRGB's RGB guide](#) - This focuses more on the aspect of connecting your physical analogue consoles to modern displays, but it covers much of the terminology used throughout this guide and explains particular details about analogue standards.

Foreword

This Guide would not have been possible without the following people to name a few:

- jfroco's work to output Batocera on a CRTs.
- rtissera's knowledge, enthusiasm and willingness to add 15khz patches.
- Calamity for his knowledge, drivers, tools and GroovyMame.
- D0023R Doozer continued work at adding 15khz to the Linux kernel.
- Rion and Atari for compiling this guide.

Glossary

The field of analog signals and CRT-related standards is very complex and occasionally misleading. We won't dive into every aspect in this guide but we will explain some relevant terms here (sorted in alphabetical order) to avoid confusion:

- **Active adapter** Little boxes that convert one signal to another. These use tiny computers with

processors and RAM to process the image digitally and send it out the other side as appropriate. Generally undesirable, as they can introduce lag and other artifacts. This is not considered “native”. This guide assumes that you are **NOT** using one of these (as they try to handle the entire conversion themselves).



- **BNC cables** These are standard aerial coaxial cords that can carry any analogue signal. For our purposes, they will be carrying an RGB video signal. There were not many consumer TVs that utilized this type of connection, but they were common with professional/broadcast reference monitors (PVM/BVM for short). You'll usually need to find a specific passive adapter that has the correct *amount* of BNC cables for your given PVM/BVM.



RGB Video Cable ~ SVGA 15-pin to 5 BNC



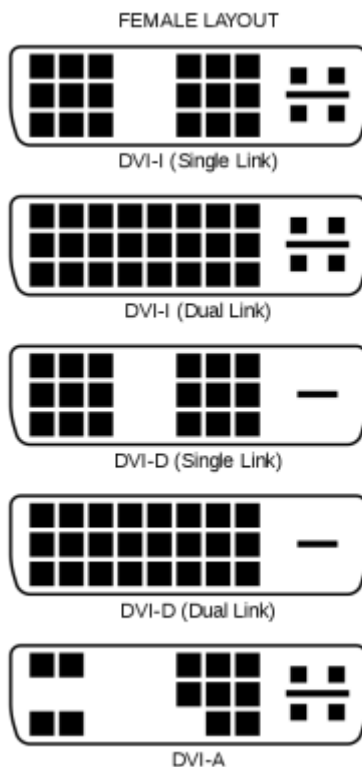
- **Component video cables** The three red blue green cable connection. Not to be confused with the similarly spelled composite cable, as it uses three cables instead of one. Not to be confused with RGB signal, as this uses a YPbPr signal. Them being colored red blue and green is purely coincidental and not related to RGB at all.



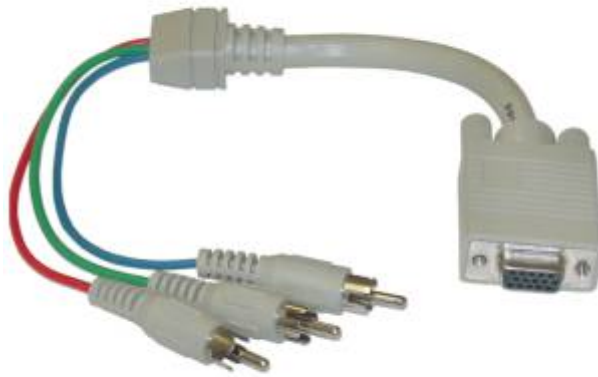
- **Composite signal** The act of sending both luminance and color channels down a single wire. Utilized by the composite video cable.
- **Composite video cable** The yellow cable. That's enough, you know what I'm talking about already.



- **DVI-I port** This is the 29 pin port in your graphics card, capable of analogue signals. You will typically find this port alongside the VGA port. Newer cards no longer include this port. It is of extreme importance to not confuse this with the identical-looking DVI-D port, which is only capable of digital signals. The DVI-I port has additional pins above and below the flat tongue, whereas DVI-D doesn't. DVI-A would also work, but that is exceptionally rare.



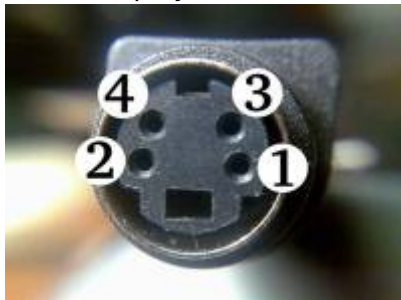
- **Passive adapter/cables** These are simple cables that adapt one interface to another. They don't involve any circuitry, all the work is done by the machine sending out the signal. Passive cables can only go analogue-to-analogue, or digital-to-digital, not analogue-to-digital or digital-to-analogue.



- **RGB video signal** In this context, refers to the analogue signal that has an individual channel for each Red, Blue and Green color signal. RGB is not compatible with component/composite/S-video by default, and a transcoder must be used to switch between the signals, but they otherwise are all analogue signals.
- **SCART** This is the 21 pin port that superseded the standard composite video output. Popular in Europe and Japan, but not many other places. Carries both analog video and audio signals. Natively uses an RGB video signal, but is backwards compatible with YPbPr, composite and S-video signals.



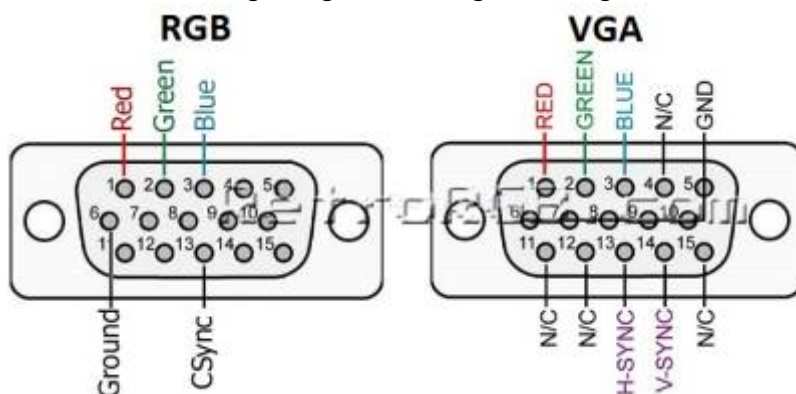
- **S-video** A slightly better version of composite video. This separated the color and video signals into two separate wires each, reducing color bleed and artifacts, but in practice the difference wasn't that significant. Common with DVD players, but not much else.



- **Transcoder** A device that can convert an RGB signal into an inferior signal like component/composite/S-video or back. Usually requires its own power source to function, but isn't an active adapter (no CPU is involved processing the image).



- **VGA port** This is the 15 pin port ubiquitous with old computer video output and classroom projectors. It is important to note the distinction between a VGA signal and the DSUB connector used to carry that signal, as not all DSUB connectors actually send a VGA signal. Refer to this console-related [article on RetroRGB](#) about it for more info. Essentially, a DSUB port is capable of sending out either an RGB or VGA signal, given the right settings.

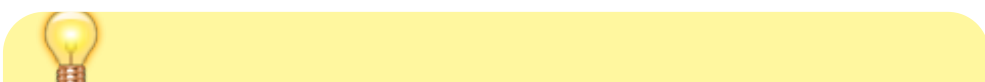


- **YPbPr video signal** A lower-bandwidth form of RGB, which has both the green and luminance channels on one wire. Used by component video cables, but can also be sent through a SCART connection.

Prerequisites

To achieve this on your setup, the following elements are needed:

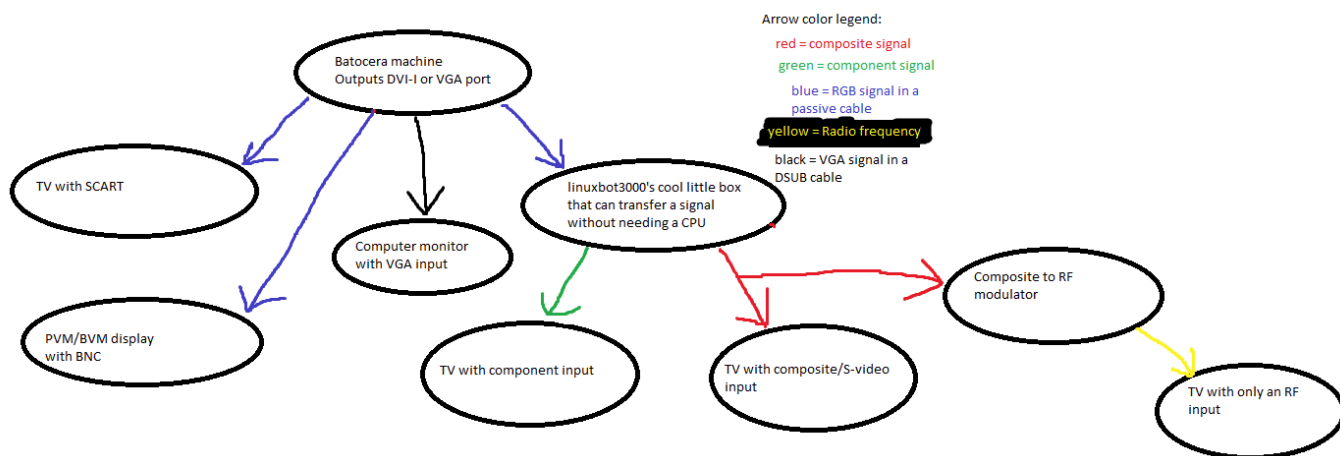
- A network connection (wired is preferred, as it needs no configuration)
- A graphics card with RGB analogue output
 - AMD/ATI (preferred) - with VGA or DVI-I port
 - Intel (has some flaws and limitations) - with VGA or DVI-I port
 - Nvidia (major flaws and limited to Super Resolutions, some older cards might work) - with VGA or DVI-I port
- If using a TV/PVM/BVM with SCART/BNC inputs, the appropriate DVI-I/VGA RGB to SCART/BNC RGB passive adapter
- If using a TV with component/composite/S-video, an appropriate DVI-I/VGA RGB to component/composite/S-video transcoder
 - If using a TV that only has RF input, you will need a composite to RF modulator in addition to the transcoder
- A way to [SSH into the Batocera machine](#) (for Windows, PuTTY works fine)
- A way to edit files over the network (for Windows, WinScp and Notepad++ work fine)



It is highly recommended using an external editor like [Notepad++](#) and [WinScp](#) for editing, as they are easier to use than command line tools and respect Unix line terminators (unlike Windows Notepad).



If any of that was confusing here's a professional-grade flowchart explaining what is required for each type of display:



Some appropriate VGA-to-component adapters:

- [RGB VGA to YPbPr Component transcoder/converter by linuxbot3000 \(ebay\) \(homepage\)](#)



- [GreenAntz RGB to component transcoder](#)



If you need composite/S-video/RF, use this transcoder:

- [RGB VGA to NTSC S-video and composite transcoder/encoder by linuxbot3000 \(ebay\) \(homepage\)](#)



Connecting the PC to the CRT

The first step is to be able to connect your Batocera computer to the CRT display itself.



During the boot process and resizing of the partition it will boot up in a non-supported resolution. Keep your TV/monitor off for the moment or on another AV channel so we don't send out **dangerous signals** to your TV/monitor. **These signals can destroy your TV.**

Also keep in mind that during the BIOS boot process the same rule applies. Have your TV/monitor off or on another input when first booting up.

To solve this we have 4 options:

- Have your TV/monitor turned off, or on another channel during the boot.
- Have a look at gambaman's excellent solution [The ultimate VGA to SCART adapter](#) over at Build Your Own Arcade Controls Forum (BYOAC).
- Use buttersoft's passthrough dongle based on gambaman's design in the link above.
- Flash you AMD/ATI card with [ATOM-15](#).

Editing the configuration files

How to edit system configuration files in Batocera

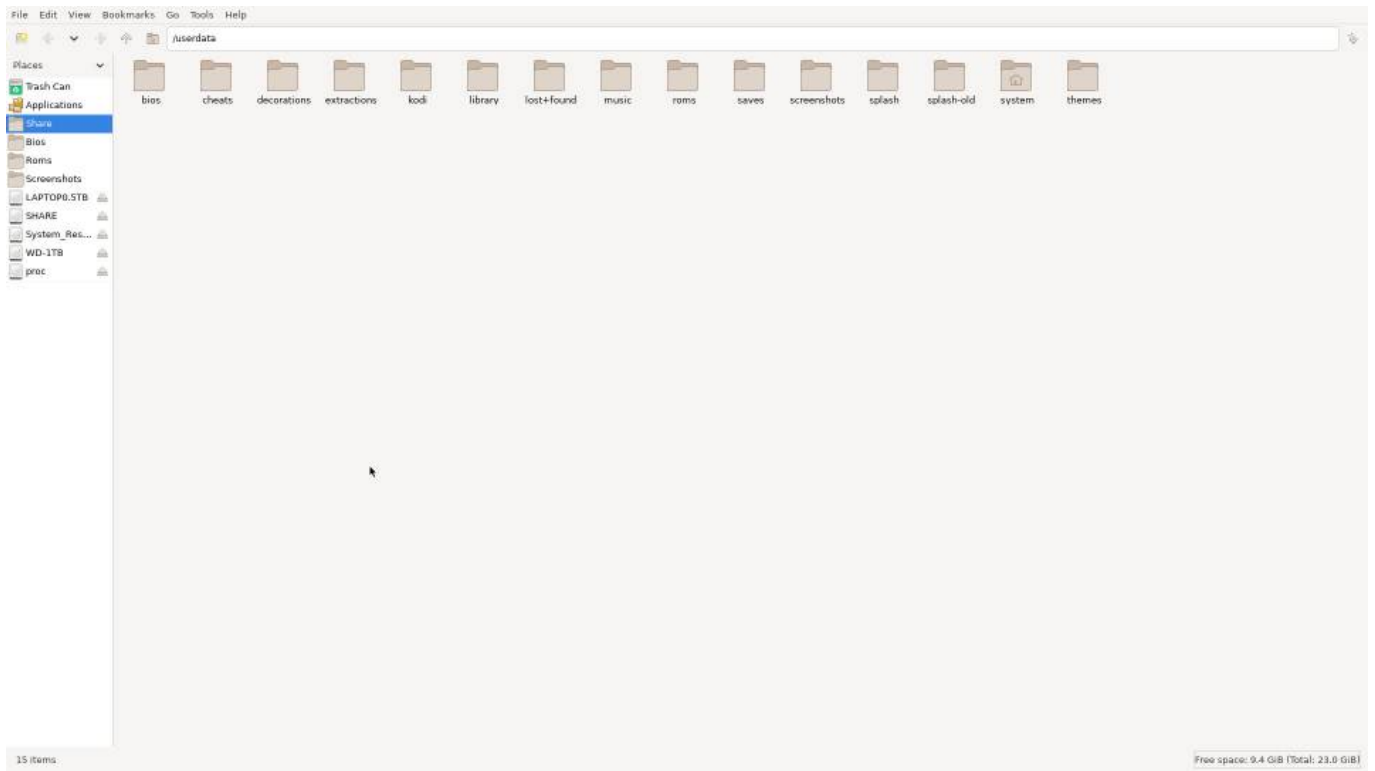
Batocera is unlike other typical Linux-based distributions as it contains most system-critical configuration files (such as files in the `/etc/` or `/usr/` folders) in its virtual filesystem, [read this for more information](#). This filesystem is expanded from the `/boot/batocera` firmware on boot into RAM, so any changes made to it by default will be forgotten on shutdown. In order to make these changes permanent, we must [save the filesystem overlay](#) (this will be mentioned in this guide whenever it is required). Overlay files are removed when updating Batocera, so they will need to be re-created again after every update (overlay files are **not** portable between different versions of Batocera).

Batocera does not share its virtual filesystem over the network share and nor is it accessible by attempting to access the drive physically from another operating system. In order to edit these files you must use Batocera itself while it is running by either utilizing its included file manager (press F1 on the systems list) or by utilizing command line tools via [SSH/xterm](#).

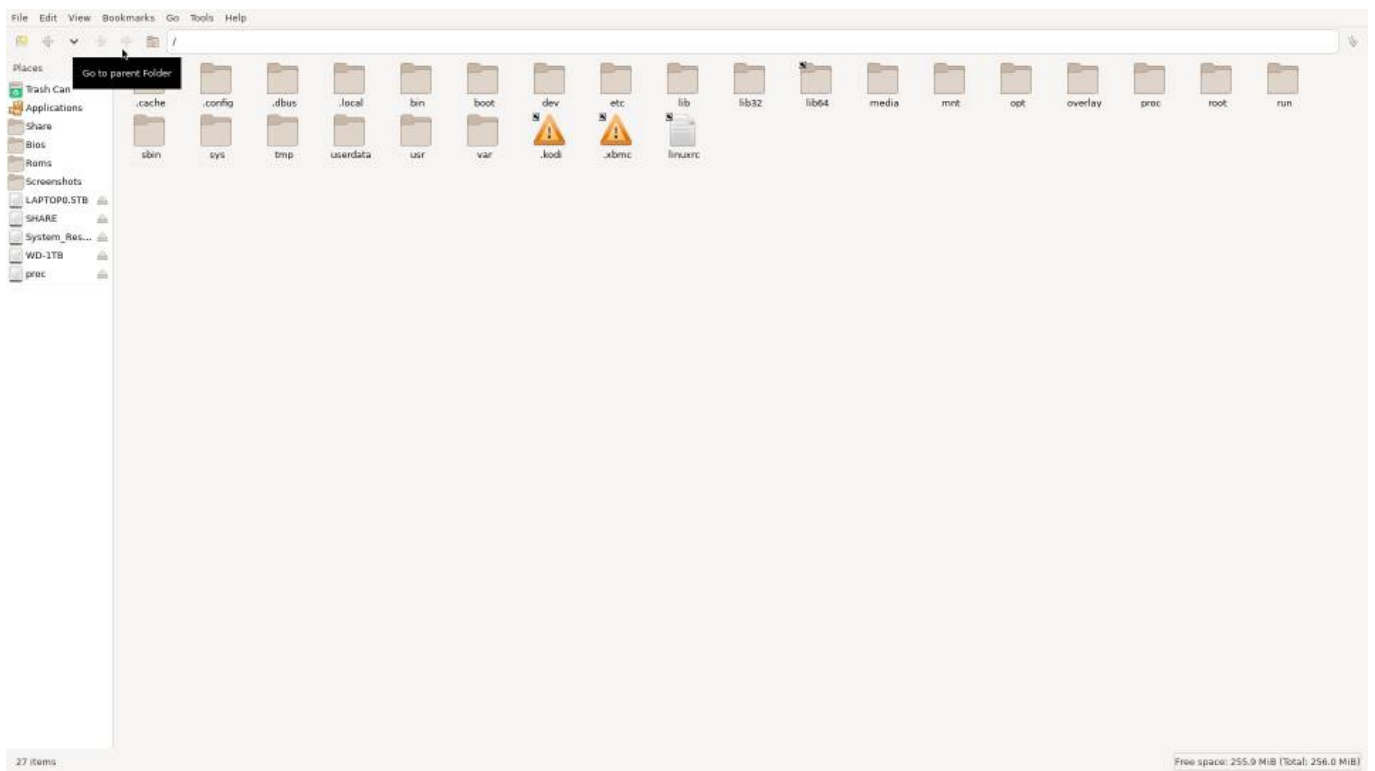
The two exceptions to this are files stored in the `/boot/` partition ([the one readable by other systems](#)) and the `/userdata/` partition ([the default location of the share folder](#)).

Using the file manager to edit files

The easiest way to manage configuration files is via the built-in file manager (x86/x86_64 only). Press F1 on the system select screen to bring up the file manager window:



Click on the “Share” folder on the left sidebar, and then click the “Up” arrow near the top to get to the root of the whole system:



You can open text files here by just double-clicking them, a basic text editing application will open them:

```

File Edit Search Options Help
# ----- A - System Options ----- #
# batocera.linux security
# enforce security
# samba password required
#system.security.enabled=0

## file system compression (btrfs only)
#system.fscompression.enabled=0

## Send cec standby command to your first tv/monitor device during shutdown
#system.cec.standby=1

## EmulationStation menu style
## default -> default all options menu
## none -> no menu except the game search menu
## bartop -> less menu, only needed for bartops
#system.es.menu=default

## Show or hide kodi in emulationstation (0,1)
kodi.enabled=0
## Start kodi at launch (0,1)
kodi.atstartup=0
## set x button shortcut (0,1)
kodi.xbutton=1

## Kodi can wait for a network component before starting
## waittime is the ip or hostname that must answer to a ping to validate the availability
## waittime is the maximum time waited when kodi boots
## If waitmode is required, kodi will not start if the component is not available
## If waitmode is wish, kodi will start if the component is not available
## If waitmode is not set or has another value, kodi will start immediately
#kodi.network.waitmode=required
#kodi.network.waittime=10
#kodi.network.waithost=192.168.0.50

## Splashscreen is enabled per default set to 0 to disable
## Set sound option to 0 to silence the video splash
#splash.screen.enabled=1
#splash.screen.sound=1

## Extend visible time of video/pictures
## Possible values are:
## outo: All the video will be played but not longer than 90s
## [0-90]: Time after ES will load in background, for RPi platform ONLY!
## [0-99]: Time after the video will be terminated, after this ES starts (recommended)
## fastboot: Stop video as soon as ES is ready to start
#splash.screen.length-fastboot
splash.screen.resize=1920x1080

## Enable video rotation (if supported) values 0,1,2,3 rotate to 0,90,180 and 270 clockwise
#splash.video.rotation=0

# ----- A1 - Platform Specific Options ----- #
## ODROID-C2
## CEC only works on your 1st tv HDMI port (EXYNOS driver limitation)

```

You can use ordinary system shortcuts like Ctrl+C to copy, Ctrl+V to paste, Alt+F4 to close an application and Alt+Tab to switch focus between applications.

Using nano to edit files via SSH

Crash course in using the nano command line text editor:

- Type nano /path/to/my/file.txt to open that file (it will show you a blank document if the file doesn't exist).
- Nano operates like a standard text editor, start typing to fill in its contents. If in SSH, you can paste the contents of your clipboard by right-clicking the terminal screen (it doesn't matter where).
- All functions are tied to Ctrl/Alt key shortcuts. The only ones you need to know are Ctrl+S to save the file, and Ctrl+X to exit.

Change the main resolution

The first step will be to change the syslinux.cfg file to be able to use Batocera at a lower resolution. To achieve this, we need to [modify the boot partition](#):

1. Connect to Batocera using SSH. See [Access Batocera Linux via ssh](#) for more information.
2. Make the boot partition writable. In the SSH session, run mount -o remount,rw /boot.

Then, we have to identify the video output we will be using:

1. Get your graphics card's analog DVI/VGA output (**without** the `card#` string) using the following command: ls /sys/class/drm/
 - This will return something like


```
card0          card0-DVI-I-1  card0-HDMI-A-2  version
card0-DP-1     card0-HDMI-A-1  renderD128
```

Here's a screenshot of it:

```
[root@BATOCERA /userdata/system]# ls /sys/class/drm/
card0          card0-DVI-I-1  card0-HDMI-A-2  version
card0-DP-1     card0-HDMI-A-1  renderD128
```

In this example our card output is **DVI-I-1**

2. Search for the syslinux file to enable booting in low resolution

 To make the next step a little bit easier for Windows users, connect to Batocera using WinScp and install Notepad++

- If legacy, the file will be at /boot/boot/syslinux/syslinux.cfg
- If UEFI, the file will be at /boot/EFI/B00T/syslinux.cfg

3. Append a space, followed by `video=[your-card-output]:640x480ieS` to the APPEND line in the file (make sure there are no additional spaces after!)

In our example:

```
APPEND label=BATOCERA console=tty3 quiet loglevel=0
vt.global_cursor_default=0 mitigations=off
```

would become

```
APPEND label=BATOCERA console=tty3 quiet loglevel=0
vt.global_cursor_default=0 mitigations=off video=DVI-I-1:640x480ieS
```

For other supported boot resolutions, see [this documentation on Github](#). Here is an example syslinux.cfg file:

[syslinux.cfg](#)

```
UI menu.c32

TIMEOUT 50
TOTALTIMEOUT 300

SAY Booting Batocera.linux...

MENU CLEAR
MENU TITLE Batocera.linux
MENU SHIFTKEY

LABEL batocera
    MENU LABEL Batocera.linux (^normal)
    MENU DEFAULT
```

```

LINUX /boot/linux
APPEND label=BATOCERA console=tty3 quiet loglevel=0
vt.global_cursor_default=0 mitigations=off video=DVI-I-1:640x480ieS
INITRD /boot/initrd.gz

LABEL verbose
MENU LABEL Batocera.linux (^verbose)
LINUX /boot/linux
APPEND label=BATOCERA vt.global_cursor_default=0
INITRD /boot/initrd.gz

```

Disabling the other video output

It is highly recommended to disable the digital outputs (HDMI/DisplayPort (DP) port) to get video exclusively out of the analog output (DVI-I/VGA port). This is not just for performance, but to avoid conflicts with switchres later on in this guide. To do so:

1. Get all the outputs on your graphics card that are capable of digital output (**with** the ``card#`` string this time) using the `ls /sys/class/drm/` command.
 - This will return something like:

```

card0          card0-DP-1      card0-DVI-I-1  version
ttm            renderD128

```

In this example, it would be `card0-DP-1` (the DisplayPort)

2. Get port name for the connector using `xrandr -display :0.0 | grep "connected"`
 - This will return something like

```

DisplayPort-0 disconnected primary (normal left inverted right x
axis y axis)
DVI-0 connected 655x500+0+0 (normal left inverted right x axis y
axis) 0mm x 0mm

```

In this example our card output is `DisplayPort-0` (notice that it is marked as disconnected)

3. Place this file into `/etc/X11/xorg.conf.d/`, replacing the outputs as appropriate:

10-monitor.conf

```

Section "Monitor"
    Identifier "[card-output]"
    Option "Ignore" "true"
EndSection

Section "Monitor"
    Identifier "[port-name]"

```

```
Option "Ignore" "true"
EndSection
```

Here is an example 10-monitor.conf file:

10-monitor.conf

```
Section "Monitor"
    Identifier "card0-DP-1"
    Option "Ignore" "true"
EndSection

Section "Monitor"
    Identifier "DisplayPort-0"
    Option "Ignore" "true"
EndSection
```

Finish by [saving the filesystem overlay](#) with the following command:

```
batocera-save-overlay
```

This will make the changes persist to the next boot (updating Batocera will remove them, however, so remember to re-apply the steps in this guide if you intend on updating).

Make 640x480i the default resolution

To enable the "640x480i" modeline to be the default xinitrc configuration:

1. Open the /etc/X11/xinit/xinitrc file.
2. Go down to

```
### radeon ###
# variable for AMD Dynamic Switchable Graphics to take amd-radeon gpu
over intel cards when such hybrid cards are available
radeon_prime="$(/usr/bin/batocera-settings-get -f /boot/batocera-
boot.conf radeon-prime)"
if test "${radeon_prime}" = "true"
then
    export DRI_PRIME=1
fi
```

3. Uncomment the three lines in that section. For example, this code:

```
#####
#####-CRT CONFIG-#####
#####
##-Default Resolution-##
```

```
#####
#xrandr -display :0.0 --newmode "640x480i" 13.10 640 664 728 832 480
484 490 525 interlace -hsync -vsync
#xrandr -display :0.0 --addmode DVI-0 "640x480i"
#xrandr -display :0.0 --output DVI-0 --mode "640x480i"
#####
#####

openbox --config-file /etc/openbox/rc.xml --startup "emulationstation-standalone"
```

would become

```
#####
#####-CRT CONFIG-#####
#####
##-Default Resolution-##
#####
xrandr -display :0.0 --newmode "640x480i" 13.10 640 664 728 832 480 484
490 525 interlace -hsync -vsync
xrandr -display :0.0 --addmode DVI-0 "640x480i"
xrandr -display :0.0 --output DVI-0 --mode "640x480i"
#####
#####

openbox --config-file /etc/openbox/rc.xml --startup "emulationstation-standalone"
```

Save the file, then make the file permanent by running batocera-save-overlay again, and reboot. You are now booting up in 640x480i mode. In **MAIN MENU** → **SYSTEM SETTINGS** → **VIDEO OUTPUT**, set your default video output to "DVI-0"/"VGA". Reboot one more time.

Disable EmulationStation from forcing 60Hz

By default, EmulationStation will try to force a refresh rate at 60 Hz at all times. This can have adverse effects on games like frame pacing (uneven scrolling), audio (crackling) and timing (speed of gameplay, physics) issues. [Here's a video of someone explaining frame pacing very slowly.](#)

We will disable ES from forcing 60 Hz at all times by uncommenting some lines.

Make a backup of /usr/bin/emulationstation-standalone. You can do so from SSH with the following:

```
cp /usr/bin/emulationstation-standalone /usr/bin/emulationstation-standalone.bak
```

Then, underneath the # try to force 60hz (specific to xorg) line, uncomment the following config lines from it:

```
# FRAMERATE="$(/usr/bin/batocera-settings-get es.framerate)"
# test -z "${FRAMERATE}" && FRAMERATE=60
# which xrandr && xrandr -r "${FRAMERATE}"
```

Your file should look like this now:

```
# try to force 60hz (specific to xorg)
FRAMERATE="$(/usr/bin/batocera-settings-get es.framerate)"
test -z "${FRAMERATE}" && FRAMERATE=60
which xrandr && xrandr -r "${FRAMERATE}"
#####
```

Don't forget to batocera-save-overlay once again. Reboot.

Managing overscan and centering in EmulationStation

As you might have noticed some of EmulationStation's menu is cut off and not centered. You may have gotten lucky and have a really well tuned TV so these steps may not be necessary, but for most TVs they will be. For this configuration you need to be able to look at your TV/monitor directly to see the changes we are going to make.



An alternative to this is to 1. use a theme optimized for CRTs that moves all elements inwards to account for overscan and 2. to disable all "crop overscan" options in emulators/cores to get a more authentic retro-game experience!

Offset (position)

We will start by aligning the top-left of the image with the top-left corner of the display. This will be important for the next step (scaling). We will do this by utilizing the following argument:

```
--screenoffset [x] [y]    Move the canvas by [x] pixels down and [y] pixels right.
```

Open userdata/system/batocera.conf and go down to this comment:

```
## Configurations generated by Batocera.linux
```

Add the following lines to right before that comment:

```
## ES Settings
es.customsargs=--screenoffset 00 00
```

Save the file. To see the changes in effect, restart EmulationStation with the following command:

```
batocera-es-swissknife --restart
```

Increase or decrease these numbers until the top-left pixel is aligned where you want it to be.

Scaling (size)

CRTs were a bit loosey-goosey with having every pixel line up 1:1 with the signal, even with a perfectly physically aligned screen you may see pixels blurring into each other. Whether you want to “fix” this is up to you. This part will simply show you how to scale your screen so you don't miss out on any content. Let's start by using the same resolution as you boot resolution. In this case, 640x480i.

Open `userdata/system/batocera.conf` again and go down to the `es.customargs` line you made earlier.

Append `--screen-size 640 480` to appear a space before `--screenoffset 00 00`:

```
## ES Settings
es.customargs="--screen-size 640 480 --screenoffset 00 00

## Configurations generated by Batocera.linux
```

`--screen-size [width] [height]` Output resolution (top-left is the anchor). Increasing will stretch the image out to the right and downwards; decreasing will squash the image from the right side toward the left and from the bottom upwards.

Remember to save the changes to `batocera.conf`. Restart EmulationStation to see the changes:

```
batocera-es-swissknife --restart
```

Persist screen scaling after exiting an emulator

We also need to make a script so this setting persists even after exiting an Emulator. Make a folder inside `/userdata/system/` called “scripts”.

Download the following file and save it to that location. Edit the `[your video port]` to your respective port if you are using a different one.

[first_script.sh](#)

```
#!/bin/bash

#Case selection for first parameter parsed
case $1 in
    gameStop)
        xrandr -display :0.0 --output [your video port] --mode
"640x480i"
        ;;
```

```
esac
```

Make the script executable with `chmod +x /userdata/system/scripts/first_script.sh`

Explanation: When the game exit it will return to the boot resolution we set in `syslinux.cfg` and also recenter the image.

Here is an example file using a DVI port for reference:

[first_script.sh](#)

```
#!/bin/bash
#This will force ES to refresh its display, grabbing the correct custom
arguments we've set.

#Set logfile location and filename
#logfile=/tmp/scriptlog.txt

#Case selection for first parameter parsed
case $1 in
#   gameStart)
#       echo "START" > $logfile
#       echo "$@" >> $logfile
#       ;;

#   gameStop)
#       xrandr -display :0.0 --output DVI-0 --mode "640x480i"
#       ;;
esac
```

Finish with `batocera-save-overlay` and reboot.

Adding Modelines (Read Resolutions) for Emulators

Right now we only have one usable modeline and that is 640x480i.

We would like to add more modelines for external emulators like for example PS2, PSP, GameCube, Wii, Wine (PC) and so on.

To do this we are going to use a program called [Switchres](#) made by Calamity. This has been integrated into Batocera as of v30. More information about Switchres can be found [here](#).

Creating the .ini file

Switchres will be activated once the appropriate `.ini` file with the correct permissions has been found at `/etc/switchres.ini`. If you intend on copying the file to this destination, you must do so

from within Batocera (the virtual filesystem cannot be accessed when Batocera isn't running).



You can download the `.ini` file required from the [Github page](#), transfer it to your Batocera machine over the network share and then use its file manager (F1) to put the file in `/etc/switchres.ini`.

If you cannot do the above, or would just like to use only SSH, do the following: First we will use nano to create a file named `switchres.ini` inside the `/etc/` folder. Run the following command:

```
nano /etc/switchres.ini
```

This will open up a new empty file for editing. Copy and paste the below into your SSH session (remember, you only need to right-click to paste):

```
#
# Switchres config
#

# Monitor preset. Sets typical monitor operational ranges:
#
# generic_15, ntsc, pal           Generic CRT standards
# arcade_15, arcade_15ex         Arcade fixed frequency
# arcade_25, arcade_31           Arcade fixed frequency
# arcade_15_25, arcade_15_25_31  Arcade multisync
# vesa_480, vesa_600, vesa_768, vesa_1024  VESA GTF
# pc_31_120, pc_70_120           PC monitor 120 Hz
# h9110, polo, pstar             Hantarex
# k7000, k7131, d9200, d9800, d9400  Wells Gardner
# m2929                           Makvision
# m3129                           Wei-Ya
# ms2930, ms929                   Nanao
# r666b                           Rodotron
#
# Special presets:
# custom   Defines a custom preset. Use in combination with crt_range0-9
options below.
# lcd      Will keep desktop's resolution but attempt variable refresh, use
in combination with lcd_range
#
    monitor           arcade_15

# Define a custom preset, use monitor custom to activate
# crt_range0-9  HfreqMin-HfreqMax, VfreqMin-VfreqMax, HFrontPorch,
HSyncPulse, HBackPorch, VfrontPorch, VSyncPulse, VBackPorch, HSyncPol,
VSyncPol, ProgressiveLinesMin, ProgressiveLinesMax, InterlacedLinesMin,
InterlacedLinesMax
# e.g.: crt_range0  15625-15750, 49.50-65.00, 2.000, 4.700, 8.000, 0.064,
```

```
0.192, 1.024, 0, 0, 192, 288, 448, 576
  crt_range0          auto
  crt_range1          auto
  crt_range2          auto
  crt_range3          auto
  crt_range4          auto
  crt_range5          auto
  crt_range6          auto
  crt_range7          auto
  crt_range8          auto
  crt_range9          auto

# Set the operational refresh range for LCD monitor, e.g. lcd_range 50-61
  lcd_range           auto

# Force a custom modeline, in XFree86 format. This option overrides the
active monitor preset configuration.
  modeline            auto

# Forces an user mode, in the format: width x height @ refresh. Here, 0 can
used as a wildcard. At least one of the three values
# must be defined. E.g. user_mode 0x240 -> SR can freely choose any width
based on the game's requested video mode, but will
# force height as 240.
  user_mode           auto

#
# Display config
#

# Select target display
# auto                Pick the default display
# 0, 1, 2, ...        Pick a display by index
# \\.\DISPLAY1, ...   Windows display name
# VGA-0, ...          X11 display name
  display             auto

# Choose a custom video backend when more than one is available.
# auto                Let Switchres decide
# adl                 Windows - AMD ADL (AMD Radeon HD 5000+)
# ati                 Windows - ATI legacy (ATI Radeon pre-HD 5000)
# powerstrip          Windows - PowerStrip (ATI, Nvidia, Matrox, etc., models up to
2012)
# xrandr              Linux - X11/Xorg
# drmkms              Linux - KMS/DRM (WIP)
  api                 auto

# [Windows] Lock video modes reported as unsupported by your monitor's EDID
  lock_unsupported_modes 1
```

```
# Lock system (non-custom) video modes, only use modes that have full
detailed timings available
    lock_system_modes          0

# Ignore video mode's refresh reported by the OS when checking ranges
    refresh_dont_care         0

# Keep changes on exit (warning: this skips video mode cleanup)
    keep_changes              0

#
# Modeline generation config
#

# Enable on-the-fly generation of video modes
    modeline_generation       1

# Allow interlaced modes (existing or generated)
    interlace                  1

# Allow doublescan modes (warning: doublescan support is broken in most
drivers)
    doublescan                 0

# Force a minimum dotclock value, in MHz, e.g. dotclock_min 25.0
    dotclock_min               0

# Maximum refresh difference, in Hz, allowed in order to synchronize. Below
this value, the mismatch does not involve penalization
    sync_refresh_tolerance    2.0

# Super resolution width: above this width, fractional scaling on the
horizontal axis is applied without penalization
    super_width                2560

# Physical aspect ratio of the target monitor. Used to compensate aspect
ratio when the target monitor is not 4:3
    aspect                     4:3

# [Experimental] Attempts to compensate consumer TVs vertical centering
issues
    v_shift_correct            0

# Calculate horizontal borders with 1-pixel precision, instead of the
default 8-pixels blocks that were required by old drivers.
# Greatly improves horizontal centering of video modes.
    pixel_precision           1

# Calculate all vertical values of interlaced modes as even numbers.
Required by AMD APU hardware on Linux
```

```
interlace_force_even    0

#
# Custom video backend config
#

# [X11] adjusts the crtc position after a new video mode is set, maintaining
the relative position of screens in a multi-monitor setup.
    screen_compositing    0

# [X11] stacks the screens vertically on startup to allow each screen to
freely resize up to the maximum width. Useful to avoid video
# glitches when using super-resolutions. screen_reordering overrides
screen_compositing.
    screen_reordering    0

# [Windows] dynamically adds new modes or updates existing ones, even on
stock AMD drivers*. This feature is experimental and is
# disabled by default. It has the following limitations and problems:
# - Synchronization is not perfect yet and the new modes may not always be
ready on time for mode switching, causing a wrong display
# output.
# - A plug-n-play audio notification will be present on startup and exit, if
the explorer shell is used.
# - Refreshing the hardware is an expensive task that takes time, specially
if the app has already entered fullscreen mode. This
# makes it unpractical for games that switch video modes more than once.
# * When used with stock AMD drivers instead of CRT Emudriver, usual
limitations apply: no support for low resolutions (below 640x480)
# nor low dotclocks.
# Not a problem however if you're using a 31 kHz monitor.
    allow_hardware_refresh    0

# Pass a custom video timing string in the native backend's format. E.g.
pstring timing for Powerstrip
    custom_timing          auto

#
# Logging
#

# Enables verbose mode (0|1)
    verbose                0

# Set verbosity level (from 0 to 3)
# 0: no messages from SR
# 1: only errors
# 2: general information
# 3: debug messages
```

```
verbosity                2
```

Press Ctrl+S to save, and then exit with Ctrl+X.

Selecting your monitor type

Switchres will need to know what kind of CRT display you have in order to generate modelines for it. Open up the `switchres.ini` file you just created with either Batocera's file manager or `nano /etc/switchres.ini` again. Scroll down to the monitor line. By default, it is:

```
monitor                  arcade_15
```


There are several types of CRT display presets to choose from. Here is just a selected few.

```
generic_15 = Generic 15.7 kHz
arcade_15 = Arcade 15.7 kHz - standard resolution
pal = PAL TV - 50 Hz/625
ntsc = NTSC TV - 60 Hz/525
arcade_15ex = Arcade 15.7-16.5 kHz - extended resolution
pc_31_120 = PC CRT 70kHz/120Hz
pc_70_120 = PC CRT 70kHz/120Hz
```

`arcade_15` works good for most standard CRTs. `generic_15` is also a good if your CRT is a bit more picky. Replace `arcade_15` with your intended preset. For example, if we were switching to `generic_15`:

```
monitor                  generic_15
```

Save the file and close it. If using `nano`, that's Ctrl+S and Ctrl+X.

 If your display device does not support one of the built-in presets, then you'll need to use the custom monitor type [here](#).

Changing the file permissions

We need to change the file access permissions. Run the following command:

```
chmod 0777 /etc/switchres.ini
```

Once again, run `batocera-save-overlay` to save the changes.

Creating a modeline

In order to create a new modeline we must first create the appropriate syntax for it. A typical modeline looks like the following:

```
xrandr -display :0.0 --newmode "320x240_60" 6.514560 320 333 364 416 240 242
245 261 -hsync -vsync
xrandr -display :0.0 --addmode "320x240_60"
```

and it will be put into the `etc/X11/xinit/xinitrc` file. Let's get started.

You can refer to [the section below](#) for a modeline to generate. In this example we are going to generate the modeline for **320x240@60hz**

```
switchres 320 240 60 -i switchres.ini -c
```

It should output a single line similar to this:

```
Switchres: Modeline "320x240_60 15.660000KHz 60.000000Hz" 6.514560 320 333
364 416 240 242 245 261 -hsync -vsync
```

We have now generated the modeline based on our monitor preset **arcade_15**. Copy this line (from after the word Modeline) and save it to a temporary text file.

```
"320x240_60 15.660000KHz 60.000000Hz" 6.514560 320 333 364 416 240 242 245
261 -hsync -vsync
```

You can opt to shorten the name by removing the `15.660000KHz 60.000000Hz` part (it is implied):

```
"320x240_60" 6.514560 320 333 364 416 240 242 245 261 -hsync -vsync
```

Switchres usage

Regular modeline generation (read the [creating a modeline section](#) above):

```
# switchres 320 240 60 -i switchres.ini -c
```

which would output:

```
Switchres: Modeline "320x240_60 15.660000KHz 60.000000Hz" 6.514560 320 333
364 416 240 242 245 261 -hsync -vsync
```

You can also skip reading the `switchres.ini` file by directly specifying the monitor in the command:

```
# switchres 320 240 60 -m generic_15 -c
```

which would output:

```
Switchres: Modeline "320x240_60 15.660000KHz 60.000000Hz" 6.514560 320 333
364 416 240 242 245 261 -hsync -vsync
```

Forced modeline generation:

```
# switchres 854 480 60 -f 854x480@60 -i switchres.ini -c
```

which would output the following:

```
Switchres: Modeline "854x480_60i 15.690000KHz 60.000000Hz" 17.415900 854 889
971 1110 480 483 489 523 interlace -hsync -vsync
```

Telling xinitrc about the new modeline(s)

Now we need to add the modeline to the xinitrc configuration. Open /etc/X11/xinit/xinitrc and scroll down to

```
#####
#####-CRT CONFIG-#####
#####
##-Default Resolution-##
#####
xrandr -display :0.0 --newmode "640x480i" 13.10 640 664 728 832 480 484 490
525 interlace -hsync -vsync
xrandr -display :0.0 --addmode DVI-0 "640x480i"
xrandr -display :0.0 --output DVI-0 --mode "640x480i"
#####
#####

openbox --config-file /etc/openbox/rc.xml --startup "emulationstation-
standalone"
```

On a new line after `xrandr -display :0.0 --output DVI-0 --mode "640x480i"`, type `xrandr -display :0.0 --newmode` and then paste in the line you copied to the temporary text file created in the previous section.

Then on another new line, type `xrandr -display :0.0 --addmode` followed by just the part enclosed in double-quotes from your modeline (including the double-quotes).

In our example the file would end up looking like this:

```
#####
#####-CRT CONFIG-#####
#####
##-Default Resolution-##
#####
xrandr -display :0.0 --newmode "640x480i" 13.10 640 664 728 832 480 484 490
525 interlace -hsync -vsync
xrandr -display :0.0 --addmode DVI-0 "640x480i"
```

```
xrandr -display :0.0 --output DVI-0 --mode "640x480i"  
#####  
#####-Modelines-#####  
#####  
xrandr -display :0.0 --newmode "320x240_60" 6.514560 320 333 364 416 240 242  
245 261 -hsync -vsync  
xrandr -display :0.0 --addmode "320x240_60"  
  
openbox --config-file /etc/openbox/rc.xml --startup "emulationstation-  
standalone"
```

Finish with `batocera-save-overlay` and reboot.

Modelines

Here are some resolutions to use

- Super Resolutions
 - 2560x240@60
 - 2560x248@58
 - 2560x256@57
 - 2560x264@55
 - 2560x272@54
 - 2560x280@52
 - 2560x288@51
 - 2560x448@60
 - 2560x464@60
 - 2560x480@60
 - 2560x496@58
 - 2560x512@57
 - 2560x544@54
 - 2560x560@52
- Emulator/Wine/PC
 - 240x240@60
 - 256x192@60
 - 288x224@60
 - 320x180@60
 - 320x200@60
 - 320x240@60
 - 320x240@60i
 - 320x256@55
 - 320x256@60
 - 352x240@60
 - 360x200@60
 - 360x240@60
 - 380x284@60
 - 384x216@60
 - 384x480@60i
 - 400x240@60

- 416×240@60
- 426×240@60
- 428×240@60
- 456×256@55
- 460×200@60
- 464@272@50
- 480×270@50
- 480×270@60
- 480×272@60
- 512×480@60i
- 640×240@60
- 640×360@60
- 640×480@60
- 640×480@60i
- 854×480@60i (4:3 from 16:9)

Configure Libretro Cores for use with Crt Switchres

Retroarch now uses [Switchres](#) instead of the old method to change its CRT resolution. Some changes have been made but for the better.

Be sure you have configured your [switchres.ini](#) before going any further in the guide.

Retroarch will not work properly if used with overlays and shaders and we don't need to use them on a CRT. We are going to disable them by default for all Emulators and set the default UI for Retroarch to Rgui.

Edit

```
/userdata/system/batocera.conf
```

Add this

```
## CRT CONFIG
global.retroarch.menu_driver=rgui
global.retroarch.menu_show_advanced_settings = "true"
global.retroarch.menu_enable_widgets = "false"
global.integerscale=0
global.smooth=0
global.rewind=0
global.shaderset=none
global.autosave=0
global.bezel_stretch=0
global.bezel=none
##
```

I have disabled both rewind and auto-save.

So if you want to have these enabled by default just leave them out.

Make a backup of

```
/userdata/system/configs/retroarch/retroarchcustom.cfg  
/userdata/system/configs/retroarch/retroarchcustom.cfg.backup
```

Edit

```
/userdata/system/configs/retroarch/retroarchcustom.cfg
```

Add

```
menu_enable_widgets = "false"
```

This will disable On-Screen Notifications and is optional

```
menu_show_advanced_settings = "true"
```

Global Libretro switchres (Same Settings for all Cores) Preferred

Configure Libretro CRT switchres in RetroArch

Choose any libretro core and game inside Batocera using your gamepad or keyboard.

Example: Core: Atari 2600 Game: H.E.R.O.

Access the menu and configure (in this order):

```
Main Menu -- Settings -- Configuration -- Save Configuration on Quit [ON]  
Main Menu -- Video -- CRT SwitchRes -- Use High Resolution Menu (Optional)  
Main Menu -- Video -- CRT SwitchRes -- CRT Super Resolution (Native)  
Main Menu -- Video -- CRT SwitchRes -- CRT SwitchRes [INI]
```

It should change to 240p/480i

Finally, save the configuration by exiting back to Emulationstation.

Per Core Override

Configure Libretro CRT switchres in RetroArch

Choose any libretro core and game inside Batocera using your gamepad or keyboard.

Example: Core: Atari 2600 Game: H.E.R.O.

Access the menu and configure (in this order):

```
Main Menu -- Settings -- Configuration -- Save Configuration on Quit [OFF]
Main Menu -- Settings -- Configuration -- Use Global Core Option Files [OFF]
Main Menu -- Video -- CRT SwitchRes -- Use High Resolution Menu (Optional)
Main Menu -- Video -- CRT SwitchRes -- CRT Super Resolution (Native)
Main Menu -- Video -- CRT SwitchRes -- CRT SwitchRes [INI]
```

It should change to 240p/480i

Finally, save the configuration for this core:

```
Quick Menu -- Overrides -- Save Core overrides
```

Do the same with the other libretro cores.

Troubleshooting

Should anything go wrong during the configuration you can always delete your Core setting for a specific core in the directory

```
/userdata/system/.config/retroarch/config/"Core_Name"
```

You can also completely remove the directory and start again (Only remove the retroarch folder)

```
/userdata/system/.config/
```

Same goes for (Only remove the retroarch folder)

```
/userdata/system/configs/
```

Or reset/delete everything using the development tool `batocera-es-swissknife` (reboot to get a clean config)

```
batocera-es-swissknife --reset-ra
```

Advanced Libretro Core and directory overrides for use with Crt Switchres

Lets' say you want to make Portable Consoles like GameBoy Advance not display an output resolution of 480i but instead want to display it in progressive scan but still keep the correct aspect ratio.

Create a Core Override file for the emulator you are going to use. In this example we are going to use mGBA.

Navigate to the directory

```
/userdata/system/.config/retroarch/config/mGBA/
```

In this folder copy you switchres.ini file and rename it to the same name as the core override file ending with .switchres.ini

```
/userdata/system/.config/retroarch/config/mGBA/mGBA.switchres.ini
```

Navigate down to the line that begins with

```
# Forces an user mode, in the format: width x height @ refresh. Here, 0 can
# used as a wildcard. At least one of the three values
# must be defined. E.g. user_mode 0x240 -> SR can freely choose any width
# based on the game's requested video mode, but will
# force height as 240.
    user_mode                auto
```

Change it to

```
user_mode                320x240
```

Change file permission

```
chmod 0777 /userdata/system/.config/retroarch/config/mGBA/mGBA.switchres.ini
```

The next step can be done via the Retroarch RGUI, but for easy of use and convenience we are going to instead edit the core override file mGBA.cfg

```
custom_viewport_height = "160"
custom_viewport_width  = "240"
video_scale_integer    = "true"
```

By doing this we are overriding switchres scaling and instead forces a resolution of 320x240 and placing the image of the perfectly scaled Gameboy Advance in a black box/window.

More information on how use Core and directory overrides can be found [here](#)

Creating you own Boot Resolution with Switchres

Let's say you have a PC Crt Monitor and like to add the boot resolution 1024x768@60hz. This can be done by using a custom **Extended Display Identification Data (EDID)**.

This will use the Monitor preset pc_31_120 **Pc Crt 31-120hz** at 1024x768@60hz

```
switchres 1024 768 60 -m pc_31_120 -e
```

The file will be named pc_31_120.bin.

- Create the folder `mkdir /lib/firmware/edid/`
- Move the file `mv pc_31_120.bin /lib/firmware/edid/`

1. Search for the syslinux file
 - If legacy, the file will be at /boot/syslinux.cfg or /boot/boot/syslinux.cfg
 - If UEFI, the file will be at /boot/EFI/syslinux.cfg or /boot/EFI/BOOT/syslinux.cfg
2. Replace the boot resolution with your edid file video=**YourCardOutput**:e
drm.edid_firmware=**YourCardOutput**:edid/pc_31_120.bin
 - In our exemple :

```
APPEND label=BATOCERA console=tty3 quiet loglevel=0  
vt.global_cursor_default=0 mitigations=off video=DVI-I-1:e  
drm.edid_firmware=DVI-I-1:edid/pc_31_120.bin
```

Configure GroovyMame

As of Batocera v30 we now have GroovyMame support.

We are now going to manually configure GroovyMame for use with a CRT.

- Go to the mame folder using the command `cd /usr/bin/mame`
- Generate the ini files needed `./mame -cc`
- this will create the following files

```
mame.ini  
plugin.ini  
ui.ini
```

(Now we need to move them)

- Create the folder `mkdir /userdata/system/.mame/`
- Move the files `mv /usr/bin/mame/*.ini /userdata/system/.mame/`

-
- Making GroovyMame's Gui display correctly on a CRT.

```
mount -o remount,rw /boot
```

- For this we need to download a Font called [uismall.bdf](#). (Place the file `uismall.bdf` you just download in to `/usr/share/fonts/TTF`)

Change file permissions

```
chmod 0777 /usr/share/fonts/TTF/uismall.bdf
```

```
batocera-save-overlay
```

Now let's configure GroovyMame.

- Edit mame.ini using nano : nano /userdata/system/.mame/mame.ini or Notepad++ for example.

- # CORE SEARCH PATH OPTIONS

```
fontpath .
```

- Change it to.

- fontpath /usr/share/fonts/TTF/

- # CORE MISC OPTIONS

- skip_gameinfo 0

- uifont default

(Change to)

- skip_gameinfo 1

- uifont uismall.bdf

- # OSD FULL SCREEN OPTIONS

- modesetting 0

(Change to)

- modesetting 1

- Edit ui.ini using nano : nano /userdata/system/.mame/ui.ini or Notepad++ for example.

- # UI OPTIONS

```
infos_text_size 0.75
```

```
font_rows 30
```

- Change it to.

- infos_text_size 1.00

```
font_rows 19
```

Let's choose Mame (GroovyMame) as the default emulator for Mame roms in Batocera.

MAIN MENU→GAMES SETTINGS→DEFAULT SETTINGS→PER SYSTEM ADVANCED CONFIGURATION→Mame

- EMULATOR: MAME
- VIDEO MODE: AUTO
- DECORATION: NONE
- STRETCH BEZELS (4K & ULTRAWIDE): NONE
- GRAPHICS BACKEND: AUTO
- BGFX BACKEND: AUTO
- BGFX VIDEO FILTER: AUTO
- CRT SWITCHRES: ON
- TATE MODE: AUTO (Change if you are rotating your TV/Monitor when playing in Vertical TATE MODE)

Advanced Miscellaneous Emulator Configuration

Dolphin - scale image to window size -

By default dolphins aspect ratio for 4:3 is set to `AspectRatio = 2` to force the aspect ratio to 4:3, but this don't actually fill the screen on a CRT and will sometimes leave black borders on the left and right sides. To solve this we need to edit the file `dolphinGenerator.py` and change the value for Aspect 4:3 to `AspectRatio = 3` to set the aspect ratio to "Stretch to window".

Navigate to

```
/usr/lib/python3.9/site-packages/configgen/generators/dolphin/
```

Edit

```
dolphinGenerator.py
```

Find the line

```
# Ratio
def getGfxRatioFromConfig(config, gameResolution):
    # 2: 4:3 ; 1: 16:9 ; 0: auto
    if "ratio" in config:
        if config["ratio"] == "4/3":
            return 2
        if config["ratio"] == "16/9":
            return 1
    return 0
```

Change to

```
# Ratio
def getGfxRatioFromConfig(config, gameResolution):
```

```
# 3: 4:3 ; 1: 16:9 ; 0: auto
if "ratio" in config:
    if config["ratio"] == "4/3":
        return 3
    if config["ratio"] == "16/9":
        return 1
return 0
```

Finish with

```
batocera-save-overlay
```

Set Core Aspect ratio to 4:3.

Dolphin - Turn off onscreen notifications -

From

```
# PanicHandlers displaymessages
dolphinSettings.set("Interface", "UsePanicHandlers", "False")
dolphinSettings.set("Interface", "OnScreenDisplayMessages", "True")
```

To

```
# PanicHandlers displaymessages
dolphinSettings.set("Interface", "UsePanicHandlers", "False")
dolphinSettings.set("Interface", "OnScreenDisplayMessages", "False")
```

Finish with

```
batocera-save-overlay
```

Libretro, turn off force notification messages

By default Batocera configgen is forcing notification messages. This can lead to unwanted effects when trying to disable them in the `retroarchcustom.cfg` file as described [here](#).

Navigate to the directory

```
/usr/lib/python3.9/site-packages/configgen/generators/libretro/
```

Edit the file `libretroConfig.py`

Find the line

```
retroarchConfig['video_font_enable'] = "true"
```

Change to

```
retroarchConfig['video_font_enable'] = '"false"'
```

Finish with

```
batocera-save-overlay
```

From:

<https://wiki.batocera.org/> - **Batocera.linux - Wiki**

Permanent link:

<https://wiki.batocera.org/batocera-and-crt?rev=1631339616>

Last update: **2021/09/11 05:53**

