Under Construction for v32

Batocera 15kHz Video Output

This guide only applies to Batocera v32. For v30/v31, refer to (insert link here).

Recommended reading: https://www.retrorgb.com/vgaconnector.html to distinguish between VGA and RGB outputs (we will be telling Batocera to output an RGB output, it does VGA by default - keep in mind this guide was made for people retrofitting their actual retro consoles to use RGB output on a modern display)

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, GroovyMame.
- D0023R Doozer continued work at adding 15khz to the Linux kernel.

Prerequisites

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

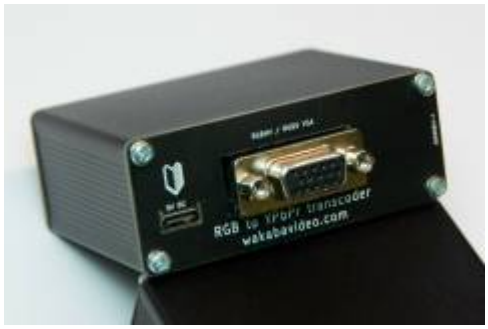
- A network connection (wired is preferred, as it needs no configuration)
- A graphics card with analog output
  - AMD/ATI (preferred) - with VGA or DVI-I connection
  - Intel (has some flaws and limitations) - with VGA or DVI-I connection
  - Nvidia (major flaws and limited to Super Resolutions, some older cards might work) - with VGA or DVI-I connection
- Analog connection to your CRT/monitor
  - If using VGA, a VGA-to-SCART/BNC (BVM/PVM)/component passive adapter/cable
  - If using DVI-I, both a DVI-I-to-VGA adapter and a VGA-to-SCART/BNC (BVM/PVM)/component passive adapter/cable
  - If you want to use composite/S-video/RF then you will need an additional RGB-to-composite 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).

Here are some recommended transcoders if you need to go all the way to composite/S-video/RF:

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

- RGB VGA to YPbPr Component transcoder/converter by linuxbot3000 (ebay) (homepage)



- GreenAntz RGB to component transcoder

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

Change the main resolution

The first step will be to change the `syslinux` 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-DP-1     card0-DVI-I-1  renderD128     ttm
     version
     ```

     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/BOOT/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 outputs (VGA/DVI-I port)**:

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  renderD128     ttm
      version
      ```

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

2. Get port name for the connector using xrandr: `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 how it is marked as currently 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).

Add Boot Modeline

Now we need to add the “640x480i” modeline

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 changes with `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”. Reboot one more time.

Disable EmulationStation from forcing 60Hz

I highly recommend using an external editor like Notepad++ and WinScp for editing

We don't want to force a vertical frequency of 60Hz all the time. This can have adverse effects on games like frame pacing (uneven scrolling), audio (crackling) and timing (speed of gameplay, physics) issues. So let's fix it by editing a few lines.

Make a backup of

```
  /usr/bin/emulationstation-standalone
```

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

From

```
  # 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}"
  ####################
```

To

```
  # 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 experience!

Scaling (size)

We are going to use `xrandr` and a function called `–scale-from wxh`. You can refer to the “`xrandr –scale-from`” section in this manual for more info.

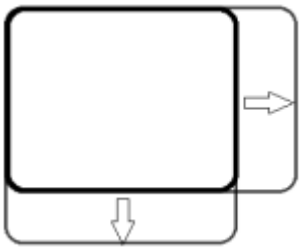
Connect via SSH and set the resolution with:

```
xrandr -display :0.0 --output [output] --scale-from 640x480
```

where `[output]` is the name of your output (eg. DVI or VGA).

Horizontal Resolution: Increasing/Decreasing this will make the image shrink or expand to the left.

Vertical Resolution: Increasing/Decreasing this will make the image shrink or expand from the bottom.



When you are satisfied with your results we need to add this setting to etc/X11/xinit/xinitrc

Edit

```
etc/X11/xinit/xinitrc
```

Add `xrandr -display :0.0 –output DVI-0 –scale-from 640×480`

Example

```
#########################
#####-CRT CONFIG-#######
#########################
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"
xrandr -display :0.0 --output DVI-0 --scale-from 640x480
######################
######################
```

We also need to make a script so this setting persists even after exiting an Emulator. Make a folder inside

```
/userdata/system/
```

called “scripts” and name it “first_script.sh”

```
/userdata/system/scripts
/userdata/system/scripts/first_script.sh
```

first_script.sh

```
#!/bin/bash
#This is an example file how Events on START or STOP can be uses
#

#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"
    xrandr -display :0.0 --output DVI-0 --scale-from "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 for reference: first_script.sh

Finish with

```
batocera-save-overlay
```

This will save the changes we made.

```
reboot
```

Advanced Fine Tuning

This will give you 1x pixel accuracy.

Let's start by using the same resolution in `xrand –scale-from` as show in the example above. In this case `640x480i`

We are going to manipulate the `Screen Size X, Y` & `Screen Position X, Y` of the Emulationstation window.

Edit

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

Go down to

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

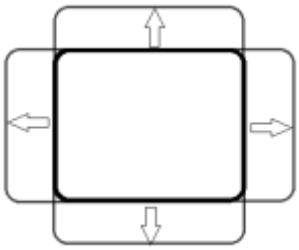
add before the line an add this

```
## ES Settings
es.customsargs=--screensize 640 480 --screenoffset 00 00

## Configurations generated by Batocera.linux
```

-screensize [width] [height] = For a canvas smaller than the full resolution.

-screenoffset [x] [y] = Move the canvas by x,y pixels.



Start by looking at you picture to see if the image needs to be bigger or smaller. Change `–screensize 640 480` to a higher or lower value and change accordingly.

Remember to save the changes to batocera.conf

Restart EmulationStation to see the changes.

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

When you are satisfied with the result do the same for Screen Position by changing the values `–screenoffset 00 00`

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 that's now apart of Batocera as of v30.

First we need to create a file named **switchres.ini** inside the folder **/usr/bin/**

Or you can download it from here: switchres.ini

More information about Switchres can be found here

From the terminal type

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

Paste this into the file

```
#
# 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
```

then end with crtrl+x to save and type Y then enter.

We need to change the file access permissions.

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

Finish with

```
batocera-save-overlay
```

This will save the changes we made.

There are several types of Monitor types to choose from. Here is just a selected few. The one we are going to use is **arcade_15** that works good for a standard crt. **generic_15** is also a good alternative.

If your display device does not match one of the built-in presets, then you'll need to use the custom monitor type here

```
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
```

In this example we are going to generate the modeline **320×240@60hz**

```
switchres 320 240 60 -i switchres.ini -c
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 the modeline and save it temporarily to a text file.

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

Lets make it easier to read it in Emulationstation and shorten the name.

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

Now we need to add the modeline

Edit

```
 etc/X11/xinit/xinitrc
```

Go 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"
```

Example

```
########################
#####-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
```

This will save the changes we made.

```
reboot
```

Modelines

Here are some resolutions to use

- Calculate Resolution using Monitor generic_15
  - `switchres 320 240 60 -m generic_15 -c`
- Calculate Resolution using setting in switchres.ini
  - `switchres 320 240 60 -i switchres.ini -c`
- Calculate Resolution using setting in switchres.ini and force the generation of the modeline
  - `switchres 854 480 60 -f 854×480@60 -i switchres.ini -c`
- Super Resolutions
  - 2560×240@60
  - 2560×248@58

    - 2560×256@57
    - 2560×264@55
    - 2560×272@54
    - 2560×280@52
    - 2560×288@51
    - 2560×448@60
    - 2560×464@60
    - 2560×480@60
    - 2560×496@58
    - 2560×512@57
    - 2560×544@54
    - 2560×560@52

- Emulator/Wine/PC
    - 240×240@60
    - 256×192@60
    - 288×224@60
    - 320×180@60
    - 320×200@60
    - 320×240@60
    - 320×240@60i
    - 320×256@55
    - 320×256@60
    - 352×240@60
    - 360×200@60
    - 360×240@60
    - 380×284@60
    - 384×216@60
    - 384×480@60i
    - 400×240@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 SwitchresSwitchres instead of the old method. 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)

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,
1920,2560,3840)
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 (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 -- 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,
1920,2560,3840)
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 complete 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 do 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 320×240 and placing the image of the perfectly scaled Gameboy Adcance 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 1024×768@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 1024×768@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
```

Libretro, Core Internal resolution up-scaling

Here is something highly unorthodox.

This will completely disable the dynamic adjustment of resolutions and horizontal frequency that switchres gives us and is highly not recommended.

With Switchres you can't use the emulators internal resolution up-scaling because that would double/triple and so on the output resolution.

But what you can do instead is to launch the emulator in a fixed resolution like 640×480 for N64 then upscale the image inside the emulator without using Switchres.

Again not recommended at all.