# LCD/LED Marquees

Under construction.

Remember back in the physical arcades how game cabinets would proudly display the logo of the game they featured on the overhead billboards? Well, thanks to the aid of modern technology, it is now possible to replicate those banners! And since we're in Batocera, we can dynamically change those artworks to reflect the game that's being played at that current moment!

## Pixelcade

The premier digital marquee manufacturers! Pixelcade has been integrated into Batocera since **v33**.


Pixelcade themselves offer installation instructions over on there website, which might update at a quicker pace than the ones on this wiki page. If the commands on this page aren't working, check that page out.

Supported platforms include:

- Raspberry Pi Zero 2
- Raspberry Pi 3
- Raspberry Pi 4
- x86_64
- Odroid N2+
- Theoretically, any aarch64/32 and/or x86_64 platform that supports Batocera **v33+**, but these have not been tested yet.

Pixelcade's software requires symlink and executable bit support in the filesystem it is installed to. This means you cannot install the software if the userdata is using FAT32, exFAT or NTFS.

Pixelcade offer their marquees in two delicious flavors. The installation instructions change depending

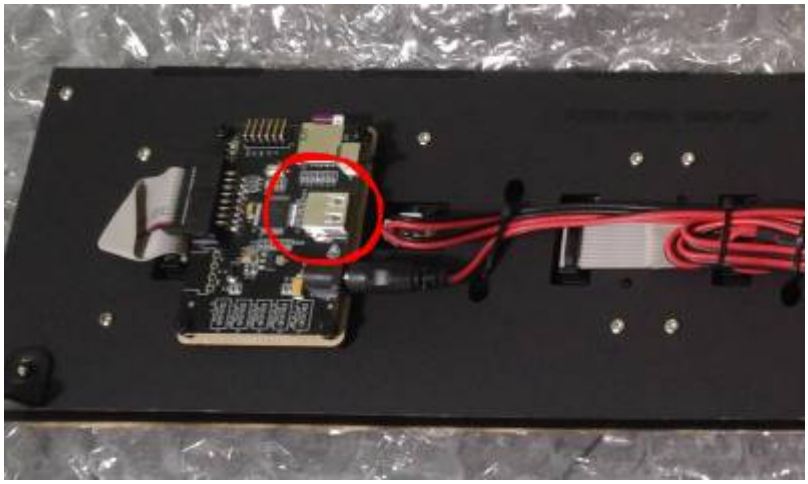
on which you want to install for.

## Pixelcade LED

This may change as Pixelcade gets better integrated into Batocera.


This is the lower resolution LED panel that's more suited to pixel-art display than full-color images. It's easy to tell this one from the LCD panel as it has a USB port on the back, while the LCD panel does not. This panel comes in two sizes, pictured above is the smaller size next to an Xbox One controller.

Installation is simple. Power on the Pixelcade panel and plug in its USB port into the Batocera machine:



Then SSH into Batocera and run the following:

```
curl -kLO -H "Cache-Control: no-cache"
https://raw.githubusercontent.com/alinke/pixelcade-linux/main/installer-
scripts/setup-batocera.sh && chmod +x setup-batocera.sh && ./setup-
batocera.sh
```

After the downloading and installation is complete, you will see the 1941 marquee being displayed on the panel. Confirm the prompt and close it.

If the panel did not display the marquee, refer to the troubleshooting section.

### Pixelcade LCD

This may change as Pixelcade gets better integrated into Batocera.

This is the higher resolution LCD panel that's suited to displaying full-color images. It supports Wi-Fi and sends its signals over the network instead of a USB cable. It is also possible to use an Ethernet connection in case a fully wired setup is preferred.

Installation uses a different command from the LED panel, so be careful. Power on the Pixelcade panel, Fix Me! (connect it to the Wi-Fi (how?)), SSH into Batocera and run the following:

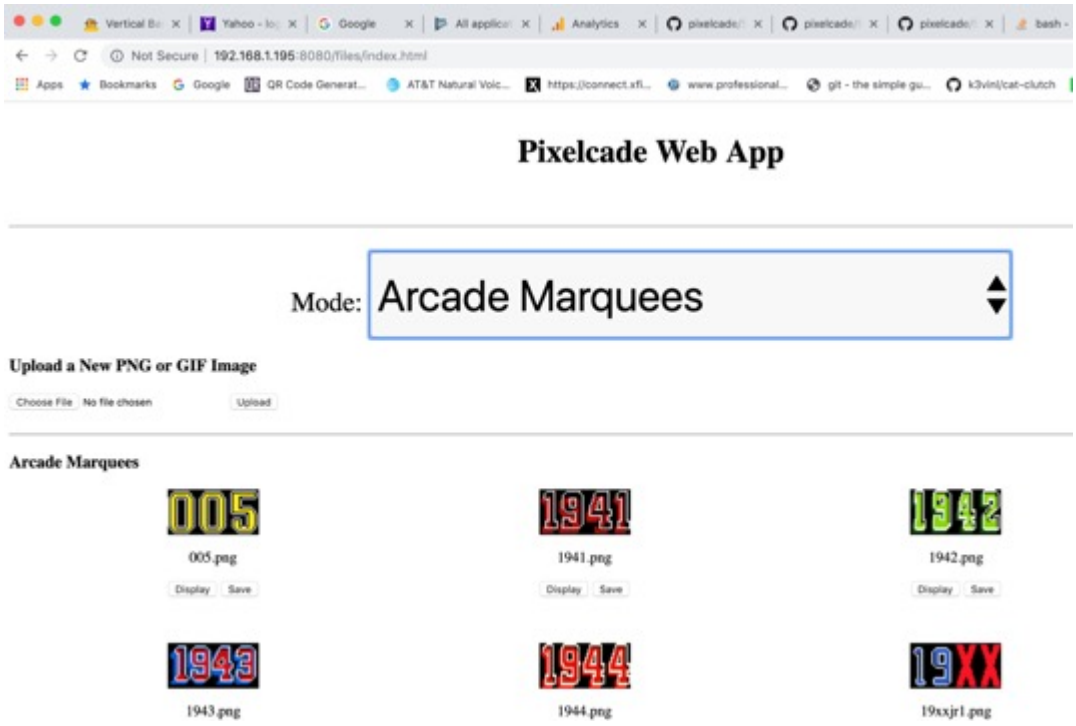
```
curl -kLO -H "Cache-Control: no-cache"
https://raw.githubusercontent.com/alinke/pixelcade-linux/main/installer-
scripts/setup-batocera-lcd.sh && chmod +x setup-batocera-lcd.sh && ./setup-
batocera-lcd.sh
```

If the panel did not display the marquee, refer to the troubleshooting section.

If you have both panels and want to use them at the same time, it is possible to install both these scripts at once.

## Pixelcade web UI

A simple web interface can be used to upload new artwork to the panel's SD card. Visit http://batocera.local:8080 (if that doesn't work, use http://<IP of Batocera>:8080) to bring it up.




## Pixelcade advanced scripting

Pixelcade has a sophisticated API that can be used to make the marquee perform certain actions. These can be programmed into the regular Batocera scripts if desired, but it might be easier to use EmulationStation's scripts instead. The ES scripts installed by Pixelcade's installer can be found in `/userdata/system/configs/emulationstation/scripts`.

For instance, here's a simplified routine that displays the game's marquee when a game is launched by Batocera (if it were installed into the `/userdata/system/configs/emulationstation/scripts/game-start` folder and marked as executable):

pixelcade.sh

```
#!/bin/bash

# Save the arguments into variables.
system="${1}"
rom="${2}"
romname="${3}"

# Convert an argument into another value.
if [[ "${system}" == "fbneo" ]]; then
        system="mame"
fi
```

```
# Switch case for certain systems.
case ${system} in
  fbneo)
    system="mame"
    ;;
  scummvm)
    rom="${rom%.*}"
    ;;
esac

# Execute this part every time this event triggers.
curl -G \
        --data-urlencode "t=${romname}" \
        http://127.0.0.1:8080/arcade/stream/${system}/`basename ${rom}`
```

## Using the two HDMI outputs on a RPi4

It is possible to use the two HDMI outputs on an RPi4, with one displaying Batocera and the other displaying the dynamic marquee. Link to the original forum post explaining this. Essentially, the Raspberry Pi draws images/videos directly to the framebuffer, which is still visible on the other display even though it's hidden by Batocera on the main display.

Artwork needs to be sourced and placed in the appropriate `Marquee` and `roms/Marquee` folders first.

Place `game.sh` into `system/configs/emulationstation/scripts/game-selected`

game.sh

```
#!/bin/bash
System=$1 #system name
Romname=${2%.*} #romname
rom=${Romname##*/}
/userdata/marquee.sh Gameselected $System "$rom"
```

Place `system.sh` into `system/configs/emulationstation/scripts/system-selected`

system.sh

```
#!/bin/bash
System=$1 #System name
/userdata/marquee.sh Systemselected $System &
```

Place `marquee.sh` in `/userdata`

marquee.sh

```
#!/bin/bash

case $1 in
Start)
Romname=$3
Gamepath=$2
marqueeimage=$Gamepath/images/$romname-marquee.png
if [ -f "/userdata/roms/Marquee/videos/$Romname.mp4" ]
then
ffmpeg -i /userdata/roms/Marquee/videos/$Romname.mp4 -vf scale=1280:720
-sws_flags bilinear -pix_fmt rgb565le -f fbdev /dev/fb0

fi

if [ -f "/userdata/roms/Marquee/hires/$Romname.jpg" ]
then
fbv /userdata/roms/Marquee/hires/$Romname.jpg -fer
elif [ -f "$marqueeimage" ]
then
fbv $marqueeimage -fer
else
fbv /userdata/roms/mame/images/mame.png -fer
fi

;;
Gameselected)
System=$2 #system name
Romname=$3 #romname

if [ -f "/userdata/roms/Marquee/$Romname.png" ]
then
fbv /userdata/roms/Marquee/$Romname.png -fer
elif [ -f "/userdata/roms/$System/images/$Romname-marquee.png" ]
then
fbv "/userdata/roms/$System/images/$Romname-marquee.png" -fer
else
fbv /userdata/roms/Marquee/mame.png -fer
fi

;;

Systemselected)
imagepath="/userdata/roms/sysimages/$2"
if [ -f "$imagepath.png" ]
then
fbv "$imagepath.png" -fer
else
fbv /userdata/roms/mame/images/mame.png -fer
fi
```

```
;;

esac
```

Place `script.sh` in `system/scripts`

script.sh

```
#!/bin/bash

case $1 in
gameStart)

gamepath=${5%/*}
romname=${5##*/}
/userdata/marquee.sh Start $gamepath ${romname%.*} &
;;

gameStop)
killall ffmpeg
;;
esac
```

# Using two computers

Link to original forum post. Link to video demonstration.

This can be done using Batocera installed onto a device and another computer/Raspberry Pi connected to a separate display. So far, this has been tested successfully using Batocera installed on an x86_64 computer/Raspberry Pi with another Raspberry Pi acting as the marquee.

1. Install Batocera onto the main machine.
2. Install Raspbian OS on the Raspberry Pi to act as the marquee (referred to as Marquee from now on).
    1. Set this up with a static IP to make things easier.
3. On the Marquee run the following:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install ffmpeg git libjpeg9-dev libjpeg9 libpng16-16
libpng-dev
```

4. Now compile FBV on the Marquee using the following commands:

```
sudo git clone https://github.com/godspeed1989/fbv.git
cd fbv
```

```
sudo ./configure
sudo make
sudo make install
```

5. Set up the Marquee to automatically start the appropriate software on boot. Open the text file with root privileges with the following:

```
sudo nano /etc/rc.local
```

   1. Then add the following right before the `exit 0` line:

```
cd /home/pi
nc -vklp 5555 | /bin/bash &
```

      Save with `Ctrl` + S.

6. Edit the bootup file:

```
sudo nano /boot/cmdline.txt
```

   1. Change the following line from:

```
console=tty1
```

      to

```
console=tty3
```

   2. In that same file, add onto the end of the line:

```
logo.nologo vt.global_cursor_default=0
```

# Troubleshooting

## My Pixelcade LED/LCD panel isn't working!

Pixelcade takes a few more moments after ES has started before it “kicks in”.

If it's still not working, it could be that it's simply not plugged in correctly. Ensure that the USB cable on both ends is firmly secure in their ports (or if using the LCD panel, that it's paired to the same Wi-Fi network as Batocera). On the Batocera machine itself, it might be worth checking if it's functional on a different port (such as a USB 3.0 port if you were using a 2.0 port, or vice-versa).

If it's still not working, try out the following:

- If the Pixelcade software was installed before the 7th of February, 2022, a bug prevented the script from working on boot. The script at `/userdata/system/custom.sh` containing the Pixelcade instructions must be removed completely and then the installation run again. Choose “yes” to wanting to reinstall Pixelcade.
- The script can be called manually with

```
killall java && cd ~ && ./custom.sh
```

    - If the marquee starts working then the problem is that the script itself is failing to launch. Ensure that it contains `/userdata/system/jdk/bin/java -jar pixelweb.jar` at least twice in the file. If it does not, it is outdated and will not work.
    - If the marquee still does not work, then this is (likely) not the issue.
- A command can be sent to the marquee manually for testing purposes. For example:

```
~/jdk/bin/java -jar pixelcade.jar -m stream -c mame -g 1941
```

- If using the LED panel (not the LCD one), run `ls /dev` and search for `ttyACM0` or `ttyACM1`. If neither are present, then your machine has not detected the Pixelcade LED device.