Waveshare PocketTerm35 Batocera Setup Guide (Raspberry Pi 5)

The Waveshare PocketTerm35 turns a Raspberry Pi 5 into a 3.5-inch handheld retro gaming terminal — but out of the box Batocera doesn't understand its gaming keys, the Esc key won't exit games, and the M.2 board's fan is loud. This guide documents a complete, reproducible Batocera 43 setup: display overlays, RetroArch keyboard mapping for the gaming keys, hotkeys, a quiet fan curve, WiFi, and a curated ROM library for the Pi 5.

Hardware used: Raspberry Pi 5 (4GB) · Waveshare PocketTerm35 housing (640×480 DPI touchscreen, QWERTY keyboard with gaming keys) · 500GB NVMe SSD on the Waveshare PCIe TO M.2 Board (E) · Batocera Linux 43.1 "Glasswing".

PocketTerm35 display setup in Batocera (config.txt)

The 3.5" display uses Waveshare's custom DPI overlays. These `.dtbo` files are **not part of Batocera** — they come from Waveshare's `3.5HDMI_E_DTBO.zip` (software guide) and must be extracted to `/boot/overlays/`. Then in `/boot/config.txt`:

```
dtparam=i2c_arm=on
dtoverlay=waveshare-35dpi-4b
dtoverlay=waveshare-35dpi-5b
dtoverlay=dwc2,dr_mode=host      # required for the keyboard (USB OTG host)
dtparam=pciex1_gen=3             # NVMe at gen 3 speed
```

In `batocera.conf`, keep EmulationStation at panel resolution: `es.resolution=max-640x480`.

Warning: a Batocera version upgrade rewrites the boot partition and deletes the custom overlays — the screen stays black afterwards. Keep backups of `waveshare-35dpi-4b.dtbo`, `waveshare-35dpi-5b.dtbo` and `config.txt` in `/userdata/system/` (survives upgrades) and copy them back after upgrading (`mount -o remount,rw /boot`).

Fix: PocketTerm35 gaming keys not working in games (RetroArch keyboard mapping)

The PocketTerm35 keyboard enumerates as a USB HID keyboard ("My Custom Pico Keyboard"). Its gaming keys emit **ordinary keyboard keycodes matching their printed labels** (verified by capturing `/dev/input/event0`):

Physical key	Keycode sent
D-pad ▲▼◀▶	arrow keys
A / B / X / Y	letters `a` / `b` / `x` / `y`
L / R	letters `l` / `r`
Select (PrtSc)	`KEY_SYSRQ` (print screen)
Start (Pause)	`KEY_PAUSE`
Esc, F1–F12	Fn layer: Fn+Q = Esc, Fn+1…0 = F1–F10

RetroArch's default keyboard layout expects retropad A on key `x`, B on `z`, Y on `a` — so the buttons do the wrong things in games. Worse, letter `r` is RetroArch's default rewind hotkey and `l` is hold-fast-forward, so the shoulder keys rewind and fast-forward the game. Fix everything in `/userdata/system/batocera.conf` (update-proof — Batocera injects these into every generated RetroArch config):

```
## POCKETTERM35 KEYBOARD BLOCK START
global.retroarch.input_player1_up=up
global.retroarch.input_player1_down=down
global.retroarch.input_player1_left=left
global.retroarch.input_player1_right=right
global.retroarch.input_player1_a=a
global.retroarch.input_player1_b=b
global.retroarch.input_player1_x=x
```

```
global.retroarch.input_player1_y=y
global.retroarch.input_player1_l=l
global.retroarch.input_player1_r=r
global.retroarch.input_player1_l2=q
global.retroarch.input_player1_r2=w
global.retroarch.input_player1_start=pause
global.retroarch.input_player1_select=print_screen
# default letter hotkeys collide with the gaming keys -> move/disable
global.retroarch.input_rewind=f6
global.retroarch.input_toggle_fast_forward=f7
global.retroarch.input_hold_fast_forward=nul
global.retroarch.input_reset=nul
global.retroarch.input_frame_advance=nul
global.retroarch.input_hold_slowmotion=nul
global.retroarch.input_toggle_slowmotion=nul
# Batocera generates input_enable_hotkey=shift which forces Shift+hotkey; disable
global.retroarch.input_enable_hotkey=nul
## POCKETTERM35 KEYBOARD BLOCK END
```

N64 keyboard controls in Batocera (mupen64plus-next)

N64 needs extra care, because Batocera's default N64 emulator is the **standalone mupen64plus**, which ignores the RetroArch keyboard binds entirely (its own defaults: Enter = Start, Left-Shift = A, Left-Ctrl = B). Switch N64 to the RetroArch core and adapt the mapping for keyboard-as-analog play — append to the same `batocera.conf` block:

```
# N64 via RetroArch core so the keyboard mapping applies (standalone mupen64plus ignores it)
n64.emulator=libretro
n64.core=mupen64plus-next
n64dd.emulator=libretro
n64dd.core=mupen64plus-next
n64h.emulator=libretro
n64h.core=mupen64plus-next
# arrows drive ONLY the analog stick; the N64 d-pad mirrors C-buttons in
# games like GoldenEye, so leaving both bound makes arrow-up look upward
n64.retroarch.input_player1_l_x_minus=left
```

```
n64.retroarch.input_player1_l_x_plus=right
n64.retroarch.input_player1_l_y_minus=up
n64.retroarch.input_player1_l_y_plus=down
n64.retroarch.input_player1_up=nul
n64.retroarch.input_player1_down=nul
n64.retroarch.input_player1_left=nul
n64.retroarch.input_player1_right=nul
# C-buttons (right analog on a pad) on IJKL
n64.retroarch.input_player1_r_y_minus=i
n64.retroarch.input_player1_r_x_minus=j
n64.retroarch.input_player1_r_y_plus=k
n64.retroarch.input_player1_r_x_plus=l
# the core puts N64-A on retropad-B and N64-B on retropad-Y; keep printed labels true
n64.retroarch.input_player1_b=a
n64.retroarch.input_player1_y=b
```

(Repeat the `n64.retroarch.*` lines with the `n64h.` prefix for the ROM-hacks system.) Resulting N64 layout: arrows = analog stick, A/B = N64 A/B, Q = Z-trigger, L/R = shoulder buttons, I/J/K/L = C-buttons, Pause = Start. In GoldenEye 007: arrows walk/turn, Q shoots, R aims, J/L strafe, I/K look up/down.

PlayStation button layout

PSX runs via the `pcsx_rearmed` RetroArch core, so the global mapping applies; the four action keys land on the shape buttons by physical position:

Key	PSX button
B (bottom)	× Cross (confirm in US games)
A (right)	○ Circle
Y (left)	□ Square
X (top)	△ Triangle
Q / W	L2 / R2

Fix: Esc does not exit games in Batocera (keyboard hotkeys need

Shift)

Batocera generates `input_enable_hotkey = "shift"` into the RetroArch config, which silently requires **holding Shift together with every keyboard hotkey** — Esc alone never exits a game. The line `global.retroarch.input_enable_hotkey=nul` in the block above removes that requirement. (On the PocketTerm35, Esc itself is the Fn+Q combo, so accidental exits are not a concern.)

Batocera keyboard hotkeys on the PocketTerm35 after this config

Keys	Function
gaming keys	exactly as labeled (SNES-style layout)
Pause / PrtSc	Start / Select
Q / W	L2 / R2
Fn+Q (Esc)	quit game
Fn+1 (F1)	RetroArch menu
Fn+2 / Fn+4	save / load state
Fn+6 / Fn+7	rewind / fast-forward

Limitation: since the gaming keys emit plain letters, keyboard-centric systems (DOS, Amiga, ScummVM) see them as typed letters — play those with the keyboard itself.

Raspberry Pi 5 performance settings for Batocera (batocera.conf)

```
system.cpu.governor=performance
kodi.enabled=0            # keyboard 'x' in ES could trigger Kodi launch
global.rewind=0           # rewind costs CPU each frame
global.autosave=1         # quit = snapshot, relaunch = resume (handheld style)
# one frame less input lag on light cores:
```

```
nes.runahead=1
famicom.runahead=1
snes.runahead=1
megadrive.runahead=1
mastersystem.runahead=1
gamegear.runahead=1
gb.runahead=1
gbc.runahead=1
gba.runahead=1
pcengine.runahead=1
```

Power: the Waveshare-recommended EEPROM setting `PSU_MAX_CURRENT=5000` removes the Pi 5's 3A current limit (check with `rpi-eeprom-config`); `vcgencmd get_throttled` should report `0x0`. **Overclocking is not recommended** in this housing — the Pi already reaches ~60°C under sustained load at stock clocks.

Fix: Raspberry Pi 5 fan too loud (PCIe TO M.2 Board E fan curve)

The M.2 Board (E) cools with a 30×30×7mm 5V blower (8000 RPM, PWM + tach) on the Pi 5 fan header — audible even at 30% duty. The Pi 5 firmware fan curve can be tuned in `/boot/config.txt`: silent below 55°C, gentle midrange, full speed only near the 85°C throttle point:

```
dtparam=fan_temp0=55000,fan_temp0_hyst=5000,fan_temp0_speed=60
dtparam=fan_temp1=63000,fan_temp1_hyst=5000,fan_temp1_speed=100
dtparam=fan_temp2=70000,fan_temp2_hyst=5000,fan_temp2_speed=160
dtparam=fan_temp3=76000,fan_temp3_hyst=5000,fan_temp3_speed=255
```

Monitor with `cat /sys/class/hwmon/hwmon2/fan1_input` (RPM) and `vcgencmd measure_temp`. Drop-in replacement if the blower develops bearing rattle: a 3007 5V blower with 4-pin JST plug (e.g. GMBYLBY 3007); pinout on the Pi 5 fan header is GND, 5V, PWM, tach.

Batocera WiFi configuration

In `batocera.conf`:

```
wifi.enabled=1
wifi.ssid=<ssid>
wifi.key=<password>
```

Apply with `batocera-wifi enable` or reboot. Batocera's connman runs with `SingleConnectedTechnology=true` and prefers ethernet — WiFi connects automatically only when no cable is plugged in, and the IP changes when it switches.

Which systems run well on a Raspberry Pi 5 handheld (ROM selection)

Source library: a 1TB Batocera share image (882GB of content) — too big for the 463GB NVMe, so a curated ~390GB subset was synced. Selection principle: systems that emulate well on a Pi 5 **and** play well on a 3.5" screen with keyboard controls.

- **Included:** full PSX (CHD set), N64, Nintendo DS (the touchscreen makes it playable), Saturn, all 8/16-bit consoles and handhelds, arcade (FBNeo, MAME, Neo Geo, Naomi, Atomiswave), Sega CD, PC Engine CD, Amiga/C64/Amstrad/Atari ST (great with the physical keyboard), OpenBOR, Daphne, all small systems.
- **Excluded:** 3DS (poor emulation on Pi 5), PSP and Dreamcast (analog-stick-dependent, 267GB together), 3DO, Neo Geo CD, MSU-1 sets (duplicates).

Procedure from a WSL/Linux PC (image lives on a NAS SMB share):

```
sudo mount -t cifs //NAS/stash /mnt/stash -o username=USER,uid=$(id -u)
sudo losetup -P /dev/loop0 /mnt/stash/games/<image>.img
sudo mount /dev/loop0p2 /mnt/imgshare        # ext4 "share" partition
rsync -a --partial --info=progress2 /mnt/imgshare/bios/ root@<pi>:/userdata/bios/
rsync -a --partial --info=progress2 \
  --exclude=/3ds --exclude=/psp --exclude=/dreamcast --exclude=/snes-msu1 \
  --exclude=/msu-md --exclude=/3do --exclude=/neogeocd \
  /mnt/imgshare/roms/ root@<pi>:/userdata/roms/
```

(Batocera SSH: user `root`, default password `linux`. After the sync, refresh game lists: ES menu → Game settings → Update gamelists, `curl http://localhost:1234/reloadgames` on the device, or `batocera-es-swissknife --restart`.)

Note on covers/metadata: rsync copies each folder's entries in byte order, and uppercase game filenames sort before the lowercase `gamelist.xml`, `images/` and `videos/` — so a system's artwork arrives only after all its games. Missing covers during a partial sync are normal; refresh the game list once the folder is complete.

Backups

- On-device: `/userdata/system/pocketterm-backup/` (custom overlays + config.txt) — survives upgrades.
- On PC: a mirror of the same files plus `batocera.conf` (which contains the WiFi password in plain text — handle accordingly).

FAQ

Why do the PocketTerm35 gaming keys type letters instead of acting as a controller? The keyboard firmware sends each gaming key's printed label as a normal keyboard keycode (A sends `a`, the d-pad sends arrow keys). There is no separate gamepad mode — the fix is remapping RetroArch's player-1 keyboard binds as shown above.

How do I exit a game on the PocketTerm35? Fn+Q sends Esc. With `input_enable_hotkey=nul` set, Esc alone exits; without it Batocera requires Shift+Esc.

Does the PocketTerm35 touchscreen work in Batocera? Yes — the Goodix capacitive touchscreen works out of the box and makes Nintendo DS games playable.

Can I overclock the Raspberry Pi 5 in the PocketTerm35? Not advisable: the enclosure already reaches ~60°C under sustained load at stock 2.4GHz, so an overclock mostly buys thermal throttling and battery drain.

Which Batocera version does this apply to? Written and tested on Batocera 43.1 (May 2026) on a Raspberry Pi 5 Model B Rev 1.1.

Revision #3
Created 2026-08-22 13:20:22 UTC by Bodo
Updated 2026-08-22 14:40:44 UTC by Bodo