

GLITCH BOY

User Manual

- INTRODUCTION pg.2
- GETTING STARTED pg.3
- MENU PAGES pg.6
- GAMES AND MODES pg.7
- GLITCH ENGINE pg.10
- CV and AUDIO INPUT pg.10
- MIDI CONTROL pg.11
- BLUETOOTH pg.15
- TROUBLESHOOTING pg.15



MELTED ELECTRONICS

INTRODUCTION

The Glitch Boy is a compact device that turns a retro console into a video synthesizer and chiptune instrument.

Built by intentionally breaking an NES emulator, Glitch Boy is a tool for video artists and creatives, perfect for creating authentic 8-bit glitch aesthetics for visuals, music videos, and video art.

Load your NES or Game Boy games from a microSD card onto the Glitch Boy, then glitch them in real time with the onboard controls, or sync the corruptions to music using the audio input or MIDI to create unique visuals.

The Glitch Boy is not only a video synth but also an 8-bit audio synthesizer: you can play the console's sound chip via MIDI, turning it into an authentic chiptune instrument.

Let's explore how it works,

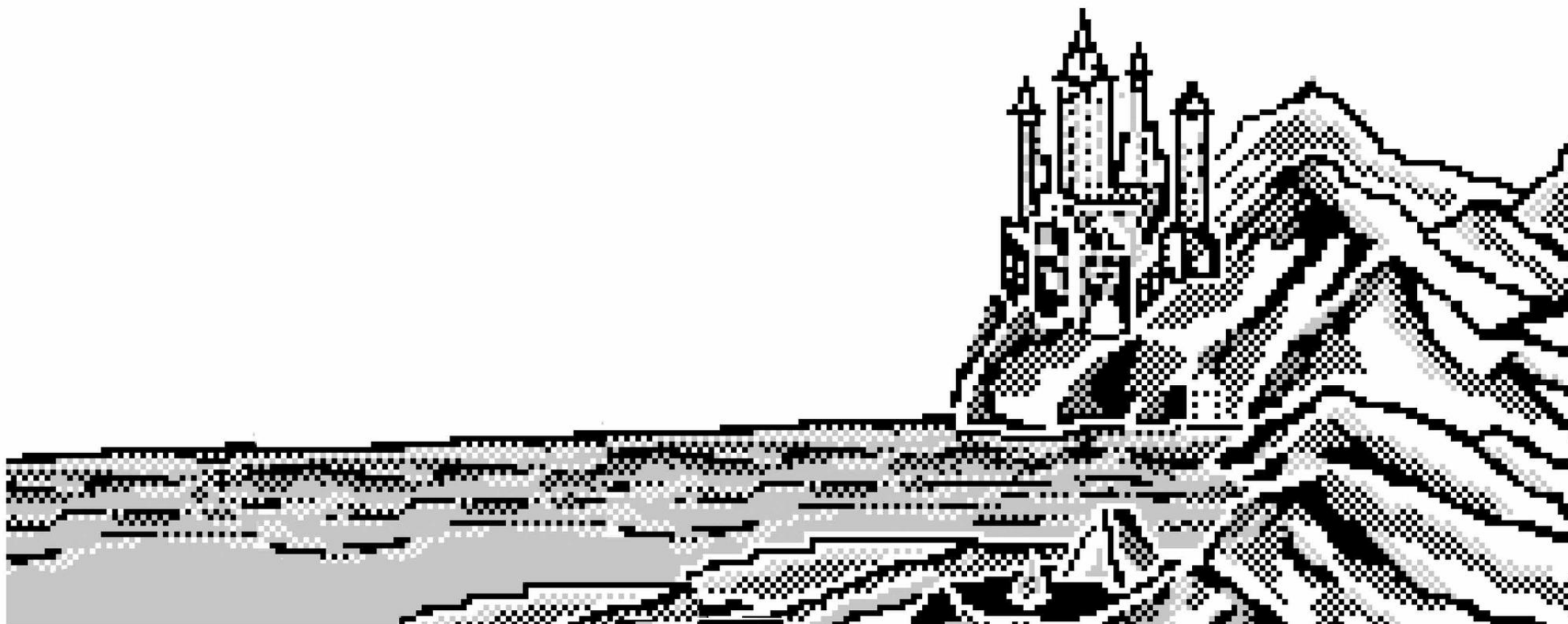
GETTING STARTED

Power-up

First, connect Glitch Boy to a display or video chain using the composite video output, labeled "Video", and/or to an audio input using the mono mini-jack output, labeled "Audio".

Insert a microSD card (32 GB max, FAT32) containing your game ROMs (.nes files; .gb and .gbc files are also supported, but support is experimental-more on that later) in a folder called "Games".

Power the device by connecting the microUSB cable to a 5v 1A power supply, or by connecting the Eurorack power cable (with the red stripe oriented down) to your Eurorack supply, and wait a few seconds for the firmware to boot. On the Games page, choose a game using Up and Down, then press A to load it.



Main panel controls

- D-pad moves in games and menus.
- A and B are used to confirm or go back in the menu and serve as game controls.
- Start sends Start in games.
- Select sends Select in games. Hold it for about half a second to open or close the menu.
- X bypasses (mutes) the current glitch setting.
- Y increases the glitch and then lets it decay, or swaps color palettes.
- Glitch A knob controls the tile glitch mode intensity.
- Glitch B knob controls the sprite/CPU/audio glitch mode intensity.

The X button behaves as a momentary killswitch. While it is pressed, the main glitch sources are muted and the game runs without corruptions. This is useful for rhythmic performance.

It can also be set to act as a latching button (glitches are toggled on or off with each button press) from the System tab of the menu.

The Y button either cycles through different color palette swaps or acts as a Glitch A control that increases when repeatedly pressed or held, then decays. It is perfect for changing the glitch character while playing a game.

Inputs

Audio In: Turns a line-level source into a control for either Glitch A or Glitch B (it can be assigned to either from the System tab). It is perfect for audio-reactive glitch visuals.

CV A and B: 0-5 V inputs for Eurorack control voltage. They can be set to drive Glitch A or B as continuous signals or act as gate inputs assigned to button presses. This lets you control the game or the X and Y button functions (see above) from your Eurorack logic.

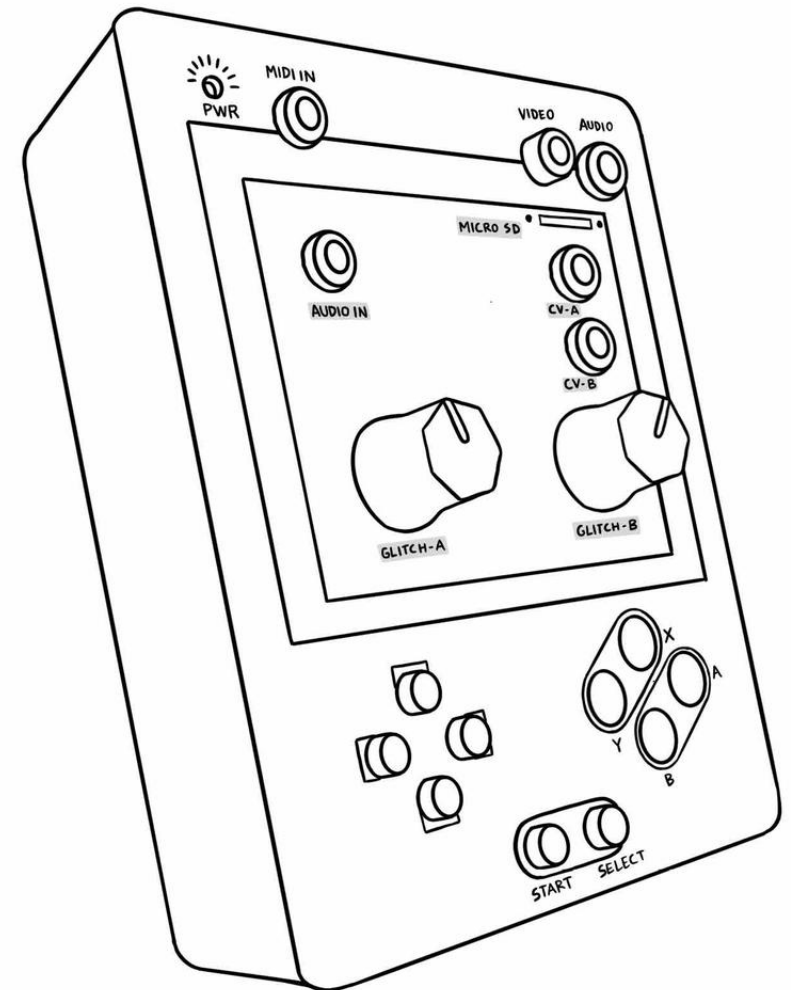
MIDI IN: This input lets you sync the glitches to MIDI clock and note-on events or control them via MIDI CC. You can also play the emulated console sound chip via MIDI when in Music mode. More on these functions later.

Opening the menu

To open the menu, hold Select for half a second.

The menu has three tabs: Games, System, and MIDI. You can navigate between them with the left and right buttons on the D-pad.

On the MIDI and System pages, Left and Right also allow you to cycle through the options on each row. In these tabs, moving above the first row gives a no-selection state. From there, Left and Right switch pages instead of cycling through the row options.



MENU PAGES

Games

Choose a game using Up and Down, then press A to load it.
Press Select on a game to show an information popup with additional ROM information.

System

The System page contains the device-wide settings.

- Mode can be Gameplay or Music. Gameplay runs the game normally.
Music turns the current core into a MIDI-controlled chip instrument.
- Core can be NES or GB for Game Boy/Game Boy Color.

NOTE: GB mode is experimental, and because it is at the edge of Glitch Boy performance, some games might not run at full speed.

- BT can be Classic or Xbox. Classic is for Bluetooth Classic controllers and Wiimotes.
- Xbox is for Xbox controllers and other BLE gamepads.
- Video can be PAL or NTSC. This chooses the composite video standard for the next boot.
- Display appears for GB. Full scales the GB picture up at the cost of some cropping at the top and bottom, while Original keeps it at its original size.
- CV A assigns the first CV input.
- CV B assigns the second CV input.
- Audio In assigns the audio-reactivity input control.
- Audio Glitch can be On or Off. When it is On, Glitch B is allowed to affect audio/APU writes.

BT, Core, and Video changes require a reboot.
GB Display, CV assignments, Audio Glitch, and Mode are applied live.

MIDI

The MIDI page is divided into three expandable areas:

- Gameplay MIDI maps MIDI notes to game-button presses.
- Video MIDI maps MIDI CCs, notes, or clock to glitch behavior.
- Audio MIDI maps MIDI notes and CCs to the NES or GB sound chip in Music mode.

Highlight a section name and press A to open or close it.
The Reset to defaults row restores the MIDI note map, glitch CCs, MIDI auto mode, and NES/GB APU MIDI voice defaults.
MIDI settings are saved when you leave the MIDI page or close the menu.

GAMES AND MODES

Gameplay mode

Gameplay is the normal mode, in which Glitch Boy acts as a video game console that you can glitch. Most of the functionality described so far belongs to this mode.
Glitch A, Glitch B, CV, MIDI CCs, MIDI auto mode, and the audio-reactive input can all affect how the image is corrupted and, when enabled, the game audio as well.

Music mode

Music mode turns the current core into a MIDI-controlled chip instrument with a built-in visualizer. After selecting it, hold Select to close the menu, and the Glitch Boy will play the notes it receives from the assigned MIDI channels, found in the Audio MIDI section. The defaults are Channels 1, 2, 3, and 10.

For NES, the four voices are:

- Pulse 1.
- Pulse 2.
- Triangle.
- Noise.

For Game Boy, the four voices are:

- Square 1.
- Square 2.
- Wave.
- Noise.

Each voice can be played independently from a different MIDI channel, assigned from the MIDI tab.
MIDI notes, velocity, CC messages, and pitch bend drive the chip voices.
The built-in visualizer replaces the game picture.
If you switch System Mode from Gameplay to Music and close the menu, the visualizer starts immediately.

While the visualizer is running:

- Press A to cycle palettes.
- Press B to cycle visualizer scenes.
- Press Y to change sprites in the scene.
- Press X to toggle the glitch on and off in the visualizer.
- Hold Select to return to the menu.

If you load a ROM while Mode is already set to Music, the game will be displayed as the picture while MIDI takes over the APU after a short boot grace period. Note that the audio might be unstable during heavy scenes or scene changes, so the visualizer is the preferred display for Music mode. To return to normal game audio, set System Mode back to Gameplay and load a game. To return to normal game audio, set System Mode back to Gameplay and close the menu.

GLITCH ENGINE

Glitch Boy has two main glitch modes.

Glitch A

Glitch A is the video/address mode. It corrupts background and tile addressing and is the main source of background and sprite tearing.

Glitch B

Glitch B is the sprite/CPU/APU mode. It can create sprite jitter, write-level corruption, and more unstable behavior. When Audio Glitch is On, this same mode can also bend audio register writes.

It can create sprite jitter, write-level corruption, and more unstable behavior. When APU Glitch is On, this same mode can also bend audio register writes.

Each glitch mode can be controlled by more than one source:

- The physical knob.
- A CV input assigned to that glitch mode.
- A MIDI CC assigned to that glitch mode.

Compatible controller X/Y buttons and CV assignments to X/Y mirror the panel glitch controls.

CV AND AUDIO-REACTIVE INPUT

CV inputs

CV lets you interface Glitch Boy with your Eurorack gear. The inputs are 0-5 V tolerant; never exceed these limits.

CV A and CV B can each be assigned from the System page.

The default behavior is:

- CV A controls Glitch A.
- CV B controls Glitch B.

When a CV input is assigned to Glitch A or Glitch B, it behaves like a continuous voltage control for that glitch mode. This lets you drive the effect with LFOs or envelopes, for instance.

CV inputs can also be assigned to any button press; they then behave like gate inputs. A high signal presses the selected button. A low signal releases it.

The gate has hysteresis, so it does not rapidly flicker at the threshold.

For X and Y (the glitch control buttons), the behavior mirrors the behavior assigned to the physical panel buttons in the System tab.

Audio input

The audio-reactive input listens to an incoming audio signal and turns its amplitude into an additional glitch control. Louder audio movement produces stronger glitch activity, and vice versa. The audio-reactive layer follows the X killswitch/toggle. If X has muted the main glitch sources, audio-reactive glitching is muted too.

Good starting patches:

- Send a drum machine into the audio input for beat-synced tearing.
- Send a click, rimshot, or modular trigger pattern for rhythmic corruption.

MIDI CONTROL

MIDI input is active by default on the MIDI input jack. Make sure you connect it using a Type A MIDI cable or converter.

MIDI has three separate jobs:

MIDI Function	What it does
Gameplay control	Turns MIDI notes into game-button presses.
Video control	Automates or triggers video glitches.
Audio control	Plays the NES or GB sound chip in Music mode.

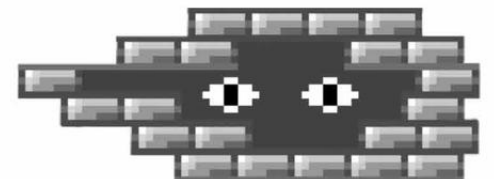
The game-button channel filter affects MIDI button emulation and glitch CCs. APU MIDI uses its own voice-channel assignments.

Gameplay MIDI

Gameplay MIDI maps MIDI notes to game buttons. The Buttons setting chooses the MIDI channel behavior:

Buttons setting	Behavior
Any	Accepts button notes on any MIDI channel.
Ch 1-Ch 16	Accepts button notes only on the selected channel.

The default is Ch 16.



Default note map:

Game button	MIDI note	Note number
Up	C4	60
Down	D4	62
Left	E4	64
Right	F4	65
A	G4	67
B	A4	69
Start	B4	71
Select	C5	72

Video MIDI

Video MIDI controls the glitch engine MIDI mapping.

Auto Mode:

Setting	Behavior
Off	No automatic random mask
Clock	Randomizes the auto mask every 24 MIDI clock pulses
Note On	Randomizes the auto mask on each incoming note-on event.

Default CC mappings:

MIDI CC	Destination	Range / Behavior
CC 1	Glitch A	Uses the full 0-127 range for the video/address mode
CC 2	Glitch B	Maps 0-127 into the Glitch B range

This lets a sequencer, DAW, Max patch, MIDI controller, or hardware instrument automate the same instability you would normally play with the knobs.



NES Audio MIDI

In NES Music mode, the four MIDI-controlled voices default to these channels:

NES voice	Default MIDI channel
Pulse 1	Channel 1
Pulse 2	Channel 2
Triangle	Channel 3
Noise	Channel 10

Each voice channel can also be set to Off or to any channel from 1 to 16.

NES Audio MIDI menu controls:

Control	Default
Pulse 1 Duty	CC 74
Pulse 2 Duty	CC 71
Pitch Bend	2 semitones
Noise Short	Off (optional)

NES performance controls:

MIDI message	Function
Velocity	Affects voice volume.
CC 7	Controls channel volume.
CC 11	Controls expression.
Pitch Bend	Bends active pitched voices.
CC 120	Sends All Sound Off.
CC 123	Sends All Notes Off.

Advanced NES APU CCs supported internally:

MIDI message	Function
CC 78	Controls envelope mode.
CC 75	Controls envelope decay.
CC 64	Controls envelope loop/hold.
CC 72	Controls length.
CC 73	Controls the triangle linear counter.
RPN 101/100 + Data Entry 6/38	Controls the pitch-bend range.

Game Boy Audio MIDI

In GB Music mode, the four MIDI-controlled voices default to these channels:

GB voice	Default MIDI channel
Square 1	Channel 1
Square 2	Channel 2
Wave	Channel 3
Noise	Channel 10

GB Audio MIDI menu controls:

Control	Default
Square 1 Duty	CC 74
Square 2 Duty	CC 71
Waveform	CC 73
Noise Short	Off (optional)
Env Mode	CC 78
Env Decay	CC 75
Env Loop	CC 64
Length CC	CC 72
Pitch Bend	2 semitones

Additional GB performance controls:

MIDI message	Function
Velocity	Affects voice volume.
CC 7	Controls volume.
CC 11	Controls expression.
Pitch Bend	Bends pitched voices.
CC 120	Sends All Sound Off.
CC 123	Sends All Notes Off.

The Waveform CC chooses between several built-in wave shapes, including saw, triangle, pulse, brighter saw, sine-like, and organ-like presets.

Saves and recent files

Glitch Boy automatically stores menu preferences internally.
Game saves are stored on the microSD card as ``.sav`` files next to the ROM.

After saving in a game, wait a moment before powering Glitch Boy off.

BLUETOOTH

Glitch Boy can receive input from several Bluetooth joysticks. The System BT setting chooses the active wireless stack.

- Classic is for Bluetooth Classic HID devices, including many keyboards and controllers.
- Xbox is for Xbox-style BLE controllers and other BLE gamepads.

Changing BT mode requires a reboot.

To pair a controller, make sure you are in the correct BT mode, then simply put it into pairing mode, and Glitch Boy will pair automatically.

Controller mapping follows the usual D-pad, A/B, Start, and Select layout.
On many controllers, Plus is Start and Minus is Select. X and Y mirror the glitch latch and hold controls when available.

TROUBLESHOOTING

If there is no video

Check the composite cable and the display input.
Make sure the display or capture device supports the selected PAL or NTSC mode.
Try a simple direct connection to a monitor or CRT before adding video mixers, converters, or capture devices.

If no games are listed

Check that the microSD card is properly inserted, in the correct orientation, and readable.
(MAX 32 GB, FAT32 format)

Check the folder name: all ROM files should be in a folder called "Games" in the root directory of the microSD card.

Check the file extensions:
- NES uses ``.nes``.
- Game Boy uses ``.gb``.
- Game Boy Color uses ``.gbc``.

If MIDI notes do not play the APU

Make sure System Mode is set to Music.
Check the MIDI/Audio MIDI voice-channel assignments.
Use CC 120 or CC 123 if you suspect stuck notes.



If MIDI buttons do not control the game

Check MIDI/Gameplay MIDI/Buttons.
If Buttons is set to a specific channel, send notes on that channel.
Check the note map.

If a CV gate is not triggering

Check the CV A or CV B assignment in System.
Use a shared ground.
Make sure the signal goes clearly high and low.

If the Bluetooth controller does not connect

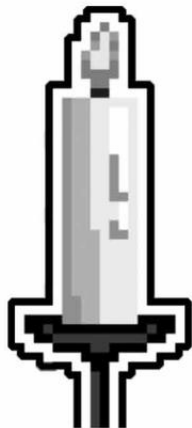
Make sure the controller is compatible and you are in the correct BT mode: use Classic for Bluetooth Classic HID devices, or use Xbox for Xbox, BLE, or HOGP devices.

If game audio is too unstable

Turn Glitch B down.
Set Audio Glitch to Off.
Check whether MIDI, CV, or the audio-reactive input is still driving Glitch B.

If saves are missing

Make sure the ROM and save files are on writable storage.
After saving in-game, wait a moment before powering off.
Check that `.sav` and `.rtc` files are being written on the micro next to the ROM when the game supports them.



THANKS TO:

Steve V., Christopher C., Aaron G., Ville K., Leo T., Arthur D., Andrew C., Matthew G., Kai M., Evan H., Matthew A., Zoe W., Antonio R., Eric H., Jesus G., Paul D., Kevin A., brainda-ta,raw, Joseph L., Matt M., Chris J., Danil I., Ronen B., Liam T., Michael K., William S., Emma K., Paul F., Felix G., Karla S., Marc F., Kevin R., Thomas A., Imma_musica, Marizu O., John S., Cody D., Mike C., Sonny B., Samuel M., Nicholas S., Ettore T., Ben M., Dany H., Pat P., Kevin S., Aaron T., Mikko N., Exsoylent G., Kris Y., Marten S., Chris L., Chan T., Mark K., Brandon S., Keith S., Leandro D., Ian S., ZigmaS, Gregory G., Joseph K., Tom, Kristen D., Brandon M., Jeffrey C., Poras D., Hannah T., Jona O., Anthony A., Christoph H., Sean M., Francesco, Michael S., Ben F., VoiceOfUnreason, Catherine S., Kyle F., John P., Cole M., Sean B., Steven L., Christopher A., Vinzvango, Joel S., John H., Mark R., Brent W., Adam R., Maxime B., Eric C., Jess K., Ryan P., Andrej C., James M., Ben, Bernardo C., Sungjae H., Jason S., Nick G., Kurt F., Jed C., Jerome D., Tyler H., Wray B., Nicholas C., Brad C., Robbie F., Grayson S., Laurelsoup, Brian K., Uglybutter, David A., Ludovic A., Domingo S., Jim W., Robert S., Riccardo S., Donald G., Cauldron of Bats, Luca C., Ryan B., Devon H., Derek S., Jt S., Adam E., Nicole B., Christian B., Ryan B., Matt H., Theron H., Michel W., Elijah R., Taylor G., John K., Jonathan F., SPIKE the Percussionist, Jack, Michael D., Iurii K., Marco L., Dougie B., Aaron Z., Alex T., Seth L., Ralf B., Jake S., Broc C., Dan, Andy M., Stephane P., Luis Z., Justin M., Jonah M., Arman B., Edwin J., Chris F., Jos D., Daniel S., Jordan H., Chris N., Angel G., Supraderk, Stefan S., Frank G., Doug W., Ra l Q., Shlomo B., Chris K., David L., StrilliVanilli, Casey L., Grixalimar, Florian S., Peter H., Jordan C., Paul A., Paul Y., Sam M., Marc M., Daniel F., FatSynthDude, Robert H., Roveen I., Oscar R., Jeremy D., Jan B., Algernon B., Joshua H., Kenta K., Derek B., Michael D., Gavin W., Alexander B., Mitchell M., Nathan G., Karl S., Mark M., Donovan L., Christopher P., Taichi M., Ash N., Ashlee M., Rodrigo T., Beau S., Joshua D., Sam G., Kristofer A., Kris M., Akifumi S., Matt V., Victor, 下方聖司, Paul H., Steven R., Charis P., Derrick B., Geoffrey P., Josh V., Isaac P., Eric P., David W., Daelan S., Robbie T., Walter V., Sylvain P., David M., Arnau, Kacy H., Joe L., Joseph D., Christopher S., Bruce E., Michael P.

for helping bring the Glitch Boy to life.



MELTED ELECTRONICS
MELTED ELECTRONICS
MELTED ELECTRONICS

Printed in June 2026
in Spazio Muffa, Torino (IT)

Melted Electronics 2026