



GNUSlashLinux

The Complete System Manual & Reference Guide

A modular, keyboard-driven Arch-based operating system
featuring `niri` WM, `Noctalia Shell`, `Helix`, `Yazi`, and high-performance CLI tools.

Alien-Tec

<https://alien-tec.com>

Contents

1	System Architecture & Philosophy	3
1.1	The Unix Philosophy	3
1.2	The Arch Linux Base	3
2	Installation & First Steps	4
2.1	Live Mode & Installer Launch	4
2.2	Initial Configuration After Booting	4
3	niri WM & Noctalia Shell Reference	5
3.1	Full Keybindings (from <code>keybinds.kdl</code>)	5
3.1.1	System, Shell & App Launcher	5
3.1.2	Window, Column & Layout Navigation	5
3.1.3	Sizing & Column/Window Formatting	6
3.1.4	Workspaces & Multi-Monitor Control	6
3.1.5	Screenshots, Audio, Display & Session Management	6
4	Kitty Terminal & Yazi File Manager	8
4.1	Kitty Terminal Reference	8
4.2	Yazi TUI File Manager Reference	8
5	Helix Text Editor (hx)	9
5.1	Basic Modes & Navigation	9
5.2	Editing, Searching & Multiple Cursors	9
5.3	The Space Menu & LSP Integration	9
6	Modern CLI Command Reference	11
6.1	File System & Navigation	11
6.2	Package Management & System Maintenance	11
7	In-House Scripts & TUI Helpers	12
7.1	System Scripts Overview	12
7.1.1	<code>git-sync</code> – Interactive Git Manager	12
7.1.2	<code>hx-open</code> – Workspace & Configuration Launcher	12
7.1.3	<code>pass-manager</code> – Password Store Migration & Backup	12
7.1.4	<code>pkexec-wl</code> – Wayland Root App Launcher	12
7.1.5	<code>smart-fire</code> – Interactive Find & Replace	12
7.1.6	<code>toggle-language</code> – System Language Configurator	13
7.1.7	<code>toggle-shell</code> – Default Shell Configurator	13
8	Backup & Disaster Recovery	14
8.1	Creating a System Snapshot	14

8.2	Restoration with Ventoy	14
9	Complete Package Overview (System Manifest)	15
9.1	System Core, Boot & Hardware	15
9.2	User Interface & Desktop (Wayland)	15
9.3	Shell & CLI Tools (Command Line)	16
9.4	Network, Bluetooth & Audio	16
9.5	Media, Documents & Applications	16
9.6	System Maintenance, Live ISO & Backup Tools	17
9.7	Development, Editors & Package Repositories	17
9.8	File Managers, Utilities & Archiving	17
10	Troubleshooting & FAQ	18
11	Important Links & Resources	19

Chapter 1

System Architecture & Philosophy

GNUSlashLinux is no ordinary Linux distribution. It was designed for power users, software developers, and system administrators who demand maximum control, uncompromising performance, and an efficient keyboard-driven workflow.

The Unix Philosophy

Traditional desktop environments (such as GNOME or KDE Plasma) often rely on resource-heavy background services, integrated monoliths, and heavyweight application suites. GNUSlashLinux radically breaks away from this approach:

- **Modularity:** Every component (window manager, status bar, terminal, file manager, editor) is an independent, highly optimized tool that communicates via standardized protocols (Wayland, IPC, CLI pipes).
- **Asynchronous CLI Toolset:** Outdated POSIX utilities have been systematically replaced by modern alternatives written in Rust (`eza`, `ripgrep`, `zoxide`, `dust`, `duf`), which natively support multithreading and caching.
- **Wayland-First with niri:** Instead of legacy X11 structures, a modern, horizontally scrolling tiling window manager is used, perfectly structuring window sessions without forcing you to deal with rigid grid layouts.

The Arch Linux Base

By utilizing Arch Linux as its foundation, GNUSlashLinux benefits from:

- **Rolling Release Model:** No major version upgrades or reinstallations required. The system always stays on the exact edge of upstream development.
- **The AUR (Arch User Repository):** Access to the world's largest community repository for Linux software.
- **Compilation Freedom:** Packages are precisely optimized for the local architecture.

Chapter 2

Installation & First Steps

GNUSlashLinux is delivered as a Live ISO, which can be used both for testing and directly for permanent installation on physical systems or virtual machines.

Live Mode & Installer Launch

After a successful boot process, the user arrives directly in the default environment, managed by `niri` and the `Noctalia Shell`.

- | | |
|----------------------------|--|
| 1. Open Terminal | Press the key combination <code>Mod</code> + <code>Enter</code> to launch the Kitty Terminal. |
| 2. Launch Installer | Enter the command <code>minstall-launcher</code> and confirm with <code>Enter</code> . |
| 3. Partitioning | Select the target drive in the graphical wizard (supports ext4, btrfs with subvolumes, and LUKS encryption). |
| 4. User Account | Define the primary user, password, and system hostname. |
| 5. Completion | Upon finishing the package installation, the installer prompts for a reboot. |

Emergency Fallback: Should the graphical installation wizard get stuck due to display server issues in the live system, the installer can be forced with root privileges and a direct Wayland context via: `pkexec-wl minstall`

Initial Configuration After Booting

After the first system boot, checking system updates via the pre-configured system alias is recommended:

```
update
```

(This alias automatically replaces the command `sudo pacman -Syu`). Afterward, monitor settings, audio routing, and design themes can be adjusted via Noctalia settings (`Mod` + `,`).

Chapter 3

niri WM & Noctalia Shell Reference

The `niri` window manager organizes windows into horizontally arranged columns (*Columns*). This means new windows logically integrate into the horizontal stream.

Full Keybindings (from `keybinds.kdl`)

System, Shell & App Launcher

<code>Mod</code> + <code>Space</code>	Toggle App Launcher (<code>noctalia msg panel-toggle launcher</code>)
<code>Mod</code> + <code>S</code>	Toggle Noctalia Control Center (<code>noctalia msg panel-toggle control-center</code>)
<code>Mod</code> + <code>,</code>	Open Noctalia Settings (<code>noctalia msg settings-toggle</code>)
<code>Mod</code> + <code>Alt</code> + <code>H</code>	Display Hotkey Overlay (shortcuts overview)
<code>Mod</code> + <code>Enter</code>	Launch Kitty Terminal
<code>Mod</code> + <code>Shift</code> + <code>Enter</code>	Open Helium Browser
<code>Mod</code> + <code>CTRL</code> + <code>Enter</code>	Open PCManFM-Qt File Manager
<code>Mod</code> + <code>E</code>	Open Helix Editor in terminal
<code>Mod</code> + <code>Y</code>	Open Yazi File Manager in terminal
<code>Mod</code> + <code>M</code>	Launch Midnight Commander in terminal
<code>Mod</code> + <code>B</code>	Launch btop++ System Monitor

Window, Column & Layout Navigation

<code>Mod</code> + <code>D</code>	Toggle Overview (<code>toggle-overview</code>)
<code>Mod</code> + <code>Shift</code> + <code>Q</code>	Close active window
<code>Mod</code> + <code>←</code> / <code>→</code> (or <code>H</code> / <code>L</code>)	Move focus to left / right column
<code>Mod</code> + <code>↑</code> / <code>↓</code> (or <code>K</code> / <code>J</code>)	Move focus up / down within column
<code>Mod</code> + <code>CTRL</code> + <code>←</code> / <code>→</code> (<code>H</code> / <code>L</code>)	Move active column left / right
<code>Mod</code> + <code>CTRL</code> + <code>↑</code> / <code>↓</code> (<code>K</code> / <code>J</code>)	Move active window up / down within column
<code>Mod</code> + <code>Home</code> / <code>End</code>	Move focus to first / last column
<code>Mod</code> + <code>CTRL</code> + <code>Home</code> / <code>End</code>	Move active column to first / last position
<code>Mod</code> + <code>[</code> / <code>]</code>	Consume / expel window in column (<code>consume-or-expel-window</code>)
<code>Mod</code> + <code>W</code>	Toggle column tabbed display (<code>toggle-column-tabbed-display</code>)

Mod + V	Toggle window floating mode
Mod + Shift + V	Switch focus between floating and tiling

Sizing & Column/Window Formatting

Mod + F	Maximize active column (<code>maximize-column</code>)
Mod + Shift + F	Toggle fullscreen mode
Mod + CTRL + F	Expand column to available width
Mod + C	Center active column
Mod + CTRL + C	Center all visible columns
Mod + R / Mod + Shift + R	Cycle / cycle backwards predefined column widths
Mod + CTRL + Shift + R	Cycle predefined window heights
Mod + CTRL + R	Reset window height
Mod + - / Mod + =	Decrease / increase column width by 10%
Mod + Shift + - / Mod + Shift + =	Decrease / increase window height by 10%

Workspaces & Multi-Monitor Control

Mod + PageDown / PageUp	Switch workspace down / up
(U/I)	
Mod + CTRL + PageDown / PageUp	Move active column to workspace down / up
(U/I)	
Mod + Shift + PageDown / PageUp	Move workspace itself down / up
(U/I)	
Mod + 1 ... 9	Switch directly to workspace 1 through 9
Mod + CTRL + 1 ... 9	Move active column to workspace 1 through 9
Mod + Shift + ←/→/↑/↓	Move focus to monitor left / right / up / down
Mod + Shift + CTRL + Arrow	Move column to monitor left / right / up / down
Mod + WheelScrollDown / Up	Switch workspace (scroll cooldown: 150ms)
Mod + CTRL + WheelScrollDown / Up	Move column to workspace down / up
Mod + WheelScrollRight / Left	Move column focus right / left
Mod + CTRL + WheelScrollRight / Left	Move column right / left

Screenshots, Audio, Display & Session Management

Print / CTRL + Print / Alt + Print	Take screenshot (selection / screen / window)
Super + Alt + L	Lock screen (<code>noctalia msg session lock</code>)
Mod + Shift + X	Open Noctalia Session Menu
Mod + Shift + E / CTRL + Alt + Del	End session / Logout
Mod + CTRL + P	Power off monitors (<code>power-off-monitors</code>)
Mod + Shift + P	Shut down system (<code>shutdown</code>)

Mod + Escape	Lock / unlock keyboard shortcuts (toggle-keyboard-shortcuts-inhibit)
XF86AudioRaiseVolume / LowerVolume / Mute	Audio Volume Up / Down / Mute
XF86MonBrightnessUp / Down	Increase / decrease display brightness

Chapter 4

Kitty Terminal & Yazi File Manager

The combination of Kitty as a GPU-accelerated terminal and Yazi as a lightning-fast TUI file manager forms the core of CLI operations.

Kitty Terminal Reference

Kitty uses the GPU to render fonts and graphics, enabling extremely smooth scrolling and direct image previews within the terminal.

CTRL + Shift + C	Copy selected text to clipboard
CTRL + Shift + V	Paste text from clipboard
CTRL + Shift + N	Open new Kitty window in current working directory
CTRL + Shift + Enter	Split Kitty window in current layout
CTRL + Shift + +	Increase terminal font size
CTRL + Shift + -	Decrease terminal font size
CTRL + Shift + O	Reset terminal font size to default
CTRL + Shift + T	Open new tab in terminal
CTRL + Shift + W	Close active tab / window
CTRL + Tab	Switch to next tab

Yazi TUI File Manager Reference

Yazi is written in Rust, built asynchronously, and provides integrated image and document previews.

h / l	Go back in directory history / Enter directory
j / k	Move cursor down / up
g g / G	Jump to top / bottom of file list
Enter	Open file with default handler
Space	Select file for batch operations
CTRL + A	Select all files in directory
y / x / p	Copy / Cut / Paste selected files
d	Move file to trash
r	Rename file or directory
f	Fuzzy search in directory via fzf
/	Incremental text filtering
z	Fast jump via zoxide history
q	Quit Yazi

Chapter 5

Helix Text Editor (hx)

Helix is a modern modal editor similar to Neovim, but unlike Neovim, it relies on a consistent **Selection** → **Action** principle and features built-in LSP (Language Server Protocol) support without cumbersome plugin configurations.

Basic Modes & Navigation

i	Switch to Insert mode before cursor
a	Switch to Insert mode after cursor
v	Switch to Visual Selection mode
Esc	Return to Normal mode / clear selection
h / j / k / l	Move cursor left / down / up / right
w / b / e	Jump word forward / backward / to end of word
f <char>	Jump to next occurrence of <char>
t <char>	Jump till before next occurrence of <char>
g g / G	Jump to first / last line of file
g h / l	Jump to start / end of line

Editing, Searching & Multiple Cursors

d	Delete current selection (Yank to register)
c	Delete selection and enter Insert mode directly
y / p	Copy (Yank) / Paste selection
u / U	Undo / Redo last change
/	Search for a term in the file
n / N	Jump to next / previous search result
*	Use current word as search term
s	Split selection by regex
C	Duplicate cursor to line below (multi-cursor)
Alt + C	Duplicate cursor to line above

The Space Menu & LSP Integration

Space + f	Project-wide file finder (uses ripgrep)
Space + b	Show and switch open buffers (files)
Space + k	LSP documentation (hover info) for symbol under cursor

<code>Space</code> + <code>r</code>	Rename variable or function project-wide (LSP Rename)
<code>Space</code> + <code>d</code>	List code errors and warnings (Diagnostics)
<code>Space</code> + <code>g</code>	View Git diff changes
<code>g</code> <code>d</code>	Jump to function / class definition (LSP)
<code>g</code> <code>y</code>	Jump to variable type
<code>:w</code> / <code>:q</code> / <code>:wq</code>	Save file / Close / Save & Close

Chapter 6

Modern CLI Command Reference

GNUSlashLinux replaces antiquated GNU commands with modern implementations. This section serves as a compact command reference for daily system usage.

File System & Navigation

<code>ls</code>	Standard listing with file type icons and color highlighting
<code>ll</code>	Detailed long format (<code>eza</code>): Shows permissions, owner, Git status
<code>lt</code>	Directory tree (<code>eza --tree</code>) up to level 2
<code>z</code> <folder>	Lightning-fast directory jumping via frequently used index (<code>zoxide</code>)
<code>zi</code>	Interactive <code>zoxide</code> selection menu via fuzzy finder
<code>rg</code> <text>	High-speed full-text search in files (<code>ripgrep</code>)
<code>df</code>	Disk usage overview as a graphical dashboard (<code>duf</code>)
<code>du</code> / <code>dut</code>	Spot disk space hogs with a bar chart (<code>dust</code>)
<code>btop</code>	Interactive system monitor for CPU, RAM, GPU, and network

Package Management & System Maintenance

Since GNUSlashLinux is based on Arch, the package manager `pacman` is used alongside AUR helpers. To simplify the workflow, the system provides pre-configured aliases:

<code>update</code>	Complete system update (<code>sudo pacman -Syu</code>)
<code>install</code> <pkg>	Install package and resolve dependencies (<code>sudo pacman -S</code>)
<code>remove</code> <pkg>	Remove package along with unneeded dependencies (<code>sudo pacman -Rns</code>)
<code>cleanup</code>	Clean package cache to free up disk space (<code>sudo pacman -Scc</code>)

Chapter 7

In-House Scripts & TUI Helpers

GNUSlashLinux provides a suite of pre-built, interactive scripts designed to simplify complex system and administrative tasks through visual TUI interfaces (`gum` and `fzf`). All scripts feature automatic bilingual language detection (German / English) and guarantee maximum security by clearing the terminal history upon exit.

System Scripts Overview

`git-sync` – Interactive Git Manager

A visual workflow assistant for Git repositories.

- **Functionality:** Checks Git status, presents an interactive file selection list, prompts for a commit message, and automatically executes a `git push` if a remote repository is present.
- **Highlights:** Uses `gum choose` with multi-selection, generates automatic timestamped commits as fallbacks, and reliably cleans up the terminal buffer.

`hx-open` – Workspace & Configuration Launcher

Quick launcher for editing system configurations and personal projects in the Helix editor.

- **Functionality:** Provides direct access to essential system files (`config.fish`, `.bashrc`, Niri WM configs) or searches projects via `fzf`, opening the chosen file directly in Helix.
- **Highlights:** Uses high-speed file search with `fd` (falling back to `find`).

`pass-manager` – Password Store Migration & Backup

A fully automated backup and migration tool for the standard GPG password manager `pass`.

- **Functionality:** Temporarily exports all encrypted passwords locally into a plaintext file for migration to a new machine. On the target system, it assists in generating a new GPG key, re-initializes `pass`, re-encrypts all entries, and securely deletes the import file using `shred`.

`pkexec-wl` – Wayland Root App Launcher

Secure launcher for running GUI applications with administrative privileges under Wayland.

- **Functionality:** Lists available desktop applications via `gum filter` and executes them post-sudo/polkit authentication with proper XWayland/Display environment variables.

`smart-fire` – Interactive Find & Replace

Interactive tool for searching and replacing text or RegEx across directory trees.

- **Functionality:** Combines `ripgrep/grep` with `fzf` previews. Allows scoping (all matches vs. filtered selection) and performs targeted `sed` replacements upon confirmation.

`toggle-language` – System Language Configurator

Management of system locale, keyboard layout, and paper sizes.

- **Functionality:** Conveniently switches the overall system between German (`de_DE.UTF-8`, keyboard `de`, paper `A4`) and English (`en_US.UTF-8`, keyboard `us`, paper `Letter`). Adjusts `localectl`, `vconsole.conf`, and Live ISO templates.

`toggle-shell` – Default Shell Configurator

Interactive switcher for the user's default login shell.

- **Functionality:** Enables seamless switching of the default shell between `bash` and `fish` via `chsh`, verifying the current status beforehand.

Chapter 8

Backup & Disaster Recovery

A productive system requires a reliable and straightforward backup strategy. GNUslashLinux relies on full system snapshots that can be created during active operation.

Creating a System Snapshot

1. Open the application launcher (**Mod** + **Space**) and launch the tool **MX Snapshot** (or the corresponding backup utility).
2. The target directory defaults to `/home/snapshot/` (alternatively, an external hard drive or partition can be chosen).
3. During operation, the tool generates a complete `.iso` backup file containing all installed packages, system configurations (*dotfiles*), user data, and settings.

Restoration with Ventoy

1. Copy the generated backup `.iso` via drag-and-drop onto a USB recovery stick prepared with **Ventoy**.
2. Boot the target machine from this USB stick.
3. Inside the live environment, open the terminal and execute the command `minstall-launcher`.
4. The entire system will be restored within minutes to the exact state in which the snapshot was created.

Chapter 9

Complete Package Overview (System Manifest)

This overview documents the key packages of this GNUSlashLinux installation, categorized by system domain, functionality, and task.

System Core, Boot & Hardware

This component provides the base of the operating system, hardware detection, and the kernel.

<code>base</code>	Essential base packages required for every GNUSlashLinux installation.
<code>linux</code>	The default Linux kernel for running the operating system.
<code>linux-firmware</code>	Driver firmware package for network cards, graphics chips, and other hardware.
<code>intel-ucode / amd-ucode</code>	Microcode updates to fix CPU bugs on Intel or AMD processors.
<code>btrfs-progs / xfsprogs</code>	Tools for creating, maintaining, and repairing Btrfs and XFS file systems.
<code>efibootmgr</code>	Management of UEFI boot entries directly from within the system.
<code>mkinitcpio-archiso</code>	Creation of the Initial RAM Disk for live systems and installation media.
<code>lshw / hwinfo</code>	Comprehensive tools for hardware detection and system analysis.

User Interface & Desktop (Wayland)

Programs responsible for graphical rendering, the window manager, and visual appearance.

<code>niri</code>	A scrollable Wayland tiling window manager acting as the main window manager.
<code>noctalia</code>	An extended desktop component or shell extension for the graphical user interface.
<code>sddm</code>	Modern display manager and login screen for graphical sessions.
<code>kitty</code>	An extremely fast, GPU-accelerated terminal emulator with rich customization options.
<code>wayland-utils</code>	/ Helper utilities and compatibility layers for Wayland and X11 applications.
<code>xwayland-satellite</code>	
<code>nwg-look</code>	Configuration tool for GTK themes in Wayland environments.

papirus-icon-theme Extensive, modern icon theme for the desktop.

Shell & CLI Tools (Command Line)

Modern and enhanced command-line tools that streamline the daily terminal workflow.

fish / fisher	An alternative, highly user-friendly shell including plugin manager.
starship	Minimalist, customizable, and high-performance shell prompt.
bttop	Modern, resource-oriented console monitor for CPU, RAM, and processes.
gdu / duf	Fast, interactive disk space analyzers.
eza / bat / fd / ripgrep	Modernized alternatives to classic Unix commands (ls , cat , find , grep).
fzf / zoxide	Fuzzy finder for files and lightning-fast directory navigation.
tmux	Terminal multiplexer for managing multiple sessions within a single window.

Network, Bluetooth & Audio

Management of network interfaces, wireless communication, and the audio system.

networkmanager	Comprehensive management of Wi-Fi, LAN, and VPN connections.
bluez / bluez-utils	Bluetooth protocol stack and control tools for wireless connections.
ufw	Simple firewall management (Uncomplicated Firewall) for system security.
pipewire-alsa pipewire-jack openssh	/ Modern, universal audio subsystem for sound and media playback. Secure remote administration and encrypted network communication via SSH.

Media, Documents & Applications

Applications for playing media, viewing documents, web browsing, and general daily use.

mpv	Powerful and modular media player for audio and video.
yt-dlp	Command-line tool for downloading videos from various platforms.
zathura	Highly modular, keyboard-driven document and PDF viewer.
pavucontrol	Graphical volume control for individual audio streams.
fontconfig ttf-nerd-fonts-symbols	/ Font management and universal symbol fonts for developer shells.
helium-browser-bin	An alternative Chromium-based web browser.

System Maintenance, Live ISO & Backup Tools

Specialized tools for system diagnostics, installation, ISO creation, and live snapshots.

<code>mx-snapshot</code>	Powerful tool for generating a live ISO / system backup of the running system.
<code>gazelle-installer</code> / <code>data-mx</code>	The installer program (including data) for setting up the system on disk.
<code>smartmontools</code>	Monitoring and diagnosis of hard drive health status (S.M.A.R.T.).
<code>rsync</code>	Classic and powerful tool for fast, incremental file backups.
<code>reflector</code>	Automatic sorting and updating of the fastest server mirrors.
<code>testdisk</code>	Data recovery tools for damaged partitions and lost file systems.
<code>tlp</code> / <code>power-profiles-daemon</code>	Power and battery optimization specifically for mobile devices.

Development, Editors & Package Repositories

Packages for software development, compilation, text editing, and package management.

<code>paru</code>	AUR helper for conveniently searching and installing third-party packages.
<code>git</code>	Distributed version control for source code and configuration files.
<code>gcc</code> / <code>make</code> / <code>patch</code>	Essential development tools and compilers for building software.
<code>python-virtualenv</code>	Creation of isolated Python environments for development projects.
<code>helix</code>	A modern, modal-based terminal text editor ("batteries-included").
<code>silicon</code>	Tool for creating beautiful graphical code snippet images.

File Managers, Utilities & Archiving

Additional file management tools, console helpers, and compression utilities.

<code>yazi</code>	A blazing-fast, terminal-based file manager in Rust with async I/O and image preview.
<code>mc</code> (Midnight Commander)	The classic, text-based dual-pane file manager for the console.
<code>pcmanfm-qt</code>	A fast, lightweight graphical file manager (Qt-based) for the desktop.
<code>xarchiver</code>	A desktop-independent, graphical archiving tool for ZIP, TAR, 7Z, etc.
<code>7zip</code> / <code>unzip</code> / <code>unrar</code>	Universal command-line tools for extracting and creating archives.
<code>jq</code>	A lightweight and flexible command-line JSON processor.
<code>wl-clipboard</code>	Clipboard tools designed specifically for Wayland environments (<code>wl-copy</code> , <code>wl-paste</code>).
<code>plocate</code>	A fast, index-based file finder (<code>locate</code> replacement).

Chapter 10

Troubleshooting & FAQ

In case unexpected issues arise during daily operation, this section provides quick solution steps.

Wayland / niri does not start	Check whether proprietary or open-source GPU drivers are loaded correctly. View log files via: <code>journalctl -b -e -u niri</code> .
No sound / Audio muted	Check PipeWire and WirePlumber status via: <code>systemctl --user status pipewire wireplumber</code> . Restart audio service: <code>systemctl --user restart wireplumber</code> .
AUR package build fails	Often the local build cache is corrupted. Clean it using: <code>yay -Syu --cleancache</code> .
Network connection disconnected	Restart NetworkManager via: <code>sudo systemctl restart NetworkManager</code> .

Chapter 11

Important Links & Resources

Official project pages, source code repositories, and vendor information:

- **Official Website:** <https://gnuslashlinux.com>
- **SourceForge Project Page:** <https://sourceforge.net/projects/gnuslashlinux/>
- **GitHub Repository:** <https://github.com/GNUSlashLinux>
- **Developer / Alien-Tec:** <https://alien-tec.com>