
目录

1 Summary	2
2 Corne 键盘配置分析 (Corne Keyboard Config Analysis)	3
2.1 概述 (Overview)	3
2.2 通用宏定义 (Common Macro Definitions)	3
系统功能宏	3
编辑与导航宏	3
2.3 标准版配置分析 (Standard Config Analysis)	3
默认层 (DEFAULT/DFT)	3
低层 (Lower Layer/LWR)	3
高层 (Raise Layer/RSE)	3
2.4 Ad-hoc v11 版配置分析 (Ad-hoc v11 Config Analysis)	4
层级扩张	4
功能分离	4
操作层 (Operation Layer/OPR)	4
2.5 核心差异对比 (Core Differences)	4
3 Corne 键盘配置对比分析: v10 vs v11 (Corne Keymap Comparison: v10 vs v11)	5
3.1 概述 (Overview)	5
3.2 生产版 v10 分析 (Analysis of Production v10)	5
3.3 Ad-hoc v11 分析与对比 (Comparison with Ad-hoc v11)	5
层级结构的急剧扩张	5
拇指键逻辑的重定义	5
操作层 (OPR) 的引入	5
3.4 结论 (Conclusion)	5

§1

Summary

§2

Corne 键盘配置分析 (Corne Keyboard Config Analysis)

§2.1

概述 (Overview)

本文详细分析了 `corne.keymap` 和 `corne_adhoc_v11.keymap` 两个配置文件的按键设置。这两个配置均基于 ZMK 固件，为 Corne 这一 42 键（或 36 键）分体键盘设计。分析重点在于层级结构 (Layer Structure)、宏定义 (Macro Definitions) 以及标准版与 Ad-hoc 版本之间的核心差异。

§2.2

通用宏定义 (Common Macro Definitions)

两个配置文件共享了一套强大的宏定义，主要针对 macOS 生态系统进行了深度优化：

系统功能宏

- `macos_screenshot (mac_sc)`: 触发 macOS 区域截图 (Cmd + Ctrl + Shift + 4)。
- `macos_screenshot_with_annotation (mac_sca)`: 触发带注释的 macOS 区域截图 (Cmd + Shift + 4)。
- `macos_input (mac_in)`: 切换输入法 (Ctrl + Alt + Space)。

编辑与导航宏

- `alt_backspace`: 删除一个词 (Alt + Backspace)。
- `ctrl_alt_left/right/up/down`: 带有组合键的导航，常用于窗口管理。
- `cmd_1` 至 `cmd_5`: 快速切换应用窗口或标签页。

§2.3

标准版配置分析 (Standard Config Analysis)

在 `corne.keymap` 中，配置相对简洁，采用了典型的三层结构：

默认层 (DEFAULT/DFT) 基于 QWERTY 布局。使用了大量的 Mod-Tap 功能，例如 `TAB` 键长按为 `LGUI`，`BSPC` 长按为 `RGUI`，`ESC` 长按为 `LCTRL`。这种设计在小键盘上极大提高了修饰键的可访问性。

低层 (Lower Layer/LWR) 主要负责数字输入 (1-0) 和方向键导航。此外，还将 `mac_sc` 和 `mac_sca` 放置在该层，方便快速截图。

高层 (Raise Layer/RSE) 集中了所有的符号键 (Symbols)，以及 RGB 灯效控制和蓝牙 (Bluetooth) 设备切换。

§2.4

Ad-hoc v11 版配置分析 (Ad-hoc v11 Config Analysis)

`corne_adhoc_v11.keymap` 展示了更复杂的演进，它被称为“Neovim Layer”配置，显然是为重度代码开发优化的：

层级扩张 该版本定义了 6 个层：DFT (0), LWR (1), RSE (2), TMP (3), FUC (4), OPR (5)。

功能分离 与标准版不同，v11 将功能控制 (RGB, BT) 独立到了 FUC (Function) 层。默认层通过 `1t` FUC RET 将回车键在长按时映射为该功能层。

操作层 (Operation Layer/OPR) 这是一个独特的创新。通过 `to OPR` 切换，该层是默认层的副本，但在拇指键上增加了 LGUI。这可能用于在特定任务（如频繁使用 GUI 快捷键）中临时改变键盘行为，而无需一直长按修饰键。

§2.5

核心差异对比 (Core Differences)

- **修饰键策略:** `corne.keymap` 右侧拇指使用 RGUI，而 v11 倾向于使用 RCTRL 和 RALT。
- **层管理:** v11 引入了 `to` 动作（切换层）而非仅仅是 `mo` 或 `1t`（临时开启层），提供了更持久的状态切换能力。
- **专业化:** v11 的命名 (Neovim Layer) 和 OPR 层的设计，体现了从通用办公向高效专业编程（尤其是 Neovim 用户）的转变。

§3 Corne 键盘配置对比分析: v10 vs v11 (Corne Keymap Comparison: v10 vs v11)

§3.1

概述 (Overview)

本文对比分析了 `corne_prod_v10.keymap` (生产版 v10) 与 `corne_adhoc_v11.keymap` (Ad-hoc v11) 的差异。这两个版本代表了从标准通用配置向高度专业化（特别是针对 Neovim 优化）配置的演进。

§3.2

生产版 v10 分析 (Analysis of Production v10)

`corne_prod_v10.keymap` 与基础的 `corne.keymap` 几乎完全一致，采用的是成熟且稳定的三层架构：

- **层级结构:** 包含 DFT (0), LWR (1), RSE (2)。
- **修饰键设计:** 使用 `mt LGUI TAB` 和 `mt RGUI BSPC` 作为左右两侧的拇指修饰键。
- **功能布局:** 数字层 (LWR) 包含导航，符号层 (RSE) 包含 RGB 和蓝牙控制。这种布局逻辑清晰，适合大多数通用办公场景。

§3.3

Ad-hoc v11 分析与对比 (Comparison with Ad-hoc v11)

`corne_adhoc_v11.keymap` 在 v10 的基础上进行了大幅度的重构，其核心差异体现在以下几个方面：

层级结构的急剧扩张 v11 将层级数量从 3 个增加到了 6 个。新增了 FUC (Function)、OPR (Operation) 和 TMP (Template) 层。这种细分允许用户将“系统控制”（如蓝牙、灯效）与“输入逻辑”（如数字、符号）完全解耦。

拇指键逻辑的重定义 在 v11 中，拇指键的逻辑变得极为复杂且强大：

- **回车键:** 在 v10 中，回车是一个单纯的按键。在 v11 中，它变成了 `1t FUC RET`，长按直接进入功能控制层。
- **修饰键切换:** v11 将右侧默认的 `RGUI` 替换为 `RCTRL`，这更符合专业程序员（尤其是 Linux/Unix 环境）的操作习惯。

操作层 (OPR) 的引入 这是 v11 最显著的特征。通过 `to OPR` 切换，该层是默认层的副本，但在拇指键上增加了 `LGUI`。这可能用于在特定任务（如频繁使用 GUI 快捷键）中临时改变键盘行为，而无需一直长按修饰键。

§3.4

结论 (Conclusion)

- **v10 (Production):** 侧重于稳定性和通用性，是经过验证的最佳实践布局。

- **v11 (Ad-hoc):** 倾重于效率和深度定制。通过增加层级深度 and 引入状态切换 (Layer Toggling)，它极大地扩展了 42 键键盘的操作空间，特别适合需要频繁切换不同快捷键模式的开发者。