

同文隨筆

version: "2020.08.08"

author: "聽雨客 <rainhear@live.com>"

同文隨筆

01. 路徑備記

- (1) TIM 已下載文件
- (2) QQ 已下載文件
- (3) 同文原版
- (4) 同文無障礙版「中文輸入法」

02. 候選數目與序號標籤

03. 方案開關設定

04. 按鍵高度設定

05. 鍵盤高度換算公式

06. 按鍵顏色示例

07. 懸浮窗橫排與豎排

08. 同文水平模式

09. 關閉或隱藏候選欄

10. 翻頁鍵設定

11. 次選與三選設定

12. 刪詞鍵設定

13. 分詞鍵設定

14. 迴車輸出注音符號

15. 指定鍵盤佈局

16. 設計衍生鍵盤

17. 快捷符號啟用

18. 反查依賴之反查翻譯器

19. 反查依賴之反查過濾器

20. 參考

01. 路徑備記

- 內置存儲絕對路徑：`/storage/emulated/0/...`
- 內置存儲相對路徑：`/sdcard/...`

(1) TIM 已下載文件

`/storage/emulated/0/Tencent/TIMfile_recv`

(2) QQ 已下載文件

① 早期版本：

`/storage/emulated/0/Tencent/QQfile_recv`

② 新近版本：

```
/storage/emulated/0/Android/data/com.tencent.mobileqq/Tencent/QQfile_recv
```

(3) 同文原版

```
/storage/emulated/0/rime
```

(4) 同文無障礙版「中文輸入法」

```
/storage/emulated/0/Android/rime
```

02. 候選數目與序號標籤

打開配置文件 `default.yaml` 或個人方案 `xxx.schema.yaml` 寫入如下代碼，可設定單葉候選項數目與序號標籤。

兩者優先級：個人方案 > 配置文件

```
# 候選菜單
menu:
  # 選單單葉候選數目
  page_size: 10
  # 替代性選詞鍵
  # 適合數字鍵參與編碼的方案單獨定義選詞鍵，無需求則無須定義
  # 如下則大寫字母為選詞鍵，即「shift + a」-「shift + j」
  alternative_select_keys: "ABCDEFGHJIJ"
  # 替代性候選標籤
  alternative_select_labels: [ ①, ②, ③, ④, ⑤, ⑥, ⑦, ⑧, ⑨, ⑩ ]
```

03. 方案開關設定

以翻月拼音 `luna_pinyin.schema.yaml` 為例，對 `switches` 等節點進行調整。

① 調整 `reset` 預設值

```
# 方案開關
switches:
  # 預設參數：0 不執行 | 1 執行
  # 中西模式，預設：0 中文 | 1 西文
  - name: ascii_mode
    reset: 0
    states: [ 中文, 西文 ]
  # 全角/半角模式，預設：0 半角 | 1 全角
  - name: full_shape
    reset: 0
    states: [ 半角, 全角 ]
  # 繁簡模式，預設：0 傳統 | 1 簡化
  - name: simplification
    reset: 0
    states: [ 漢字, 汉字 ]
  # 中西標點模式，預設：0 中文句讀 | 1 西文標點
  - name: ascii_punct
    reset: 0
    states: [ 。, , , ., ' ]
```

② 增加「臺灣正體」用字模式

```
# 1. 方案開關
switches:
  # 繁正模式，預設： 0 通用繁體 | 1 臺灣正體
  - name: trad_tw
    reset: 0
    states: [ 繁體, 正體 ]

# 2. 輸入引擎
engine:
  # 過濾器
  filters:
    - simplifier # 主用字轉換器，對應 simplification
    - simplifier@trad_tw # 副用字轉換器，對應 @trad_tw

# 3. 繁正轉換，對應 simplifier@trad_tw
trad_tw:
  # 選項名稱
  option_name: trad_tw
  # 轉換配置
  opencc_config: t2tw.json
  # 作用標識範圍，當前對應 engine/segmentors/abc_segmentor
  tags: abc
  # 是否提示轉換前文字： none 無 | char 單字 | all 所有
  tips: none
  # 是否僅將轉換結果顯示在備注中，需配合 tips 生效
  show_in_comment: false
```

③ 在 ① ② 基礎上整合「繁簡」「繁正」模式

```
# 方案開關
switches:
  # 整合型用字模式
  # 繁簡模式，預設： 0 通用繁體 | 1 臺灣正體 | 2 簡化漢字
  - options: [ noop, trad_tw, simplification ]
    reset: 1
    states: [ 繁體, 正體, 簡化 ]
```

04. 按鍵高度設定

優先級： C > B > A

A：主題風格 -> 按鍵高度節點， B 與 C 未設定時，主題預設的按鍵高度。

```
# 界面風格
style:
  key_height: 40 # 預設鍵高
```

B：鍵盤佈局 -> 按鍵高度節點，當前鍵盤佈局的**全局按鍵高度**。

為方便主題聯動設計，建議採用**錨點語法**。

```
# 高度錨點
height:
  1: &kh1 42 # 默認鍵盤、數字鍵盤、編輯鍵盤之全局按鍵高度
  2: &kh2 36 # 默認鍵盤之首行指定按鍵高度

# 鍵盤佈局
preset_keyboards:
  # 默認佈局
  default:
    height: *kh1 # 對應 &kh1 錨點
```

C：某行首顆按鍵 -> 按鍵高度屬性，可定義本行按鍵高度。

同一行僅能指定一次按鍵高度，首顆按鍵定義後，本行其餘按鍵無須再定義。

此行結束後，下一行按鍵將默認採用 **B** 中設定值。

```
# 鍵盤佈局
preset_keyboards:
  # 默認鍵盤
  default:
    # ...
    keys:
      # 第一行
      - {click: 1, long_click: '!', height: *kh2} # 對應 &kh2 錨點
      - {click: 2, long_click: '@'}
      - {click: 3, long_click: '#'}
      # ...
```

05. 鍵盤高度換算公式

A 佈局：共計 m 行按鍵，每顆按鍵高度 h_1 ，按鍵行距 h_0

B 佈局：共計 n 行按鍵，每顆按鍵高度 h_2 ，按鍵行距 h_0

C 佈局：共計 $x + y$ 行按鍵， x 行每顆按鍵高度 h_3 ， y 行每顆按鍵高度 h_4 ，按鍵行距 h_0

【公式】：

$$h_1 \times m + h_0 \times (m - 1) = h_2 \times n + h_0 \times (n - 1)$$

$$h_1 \times m + h_0 \times (m - 1) = h_3 \times x + h_4 \times y + h_0 \times (x + y - 1)$$

【等效】：

$$h_2 = \frac{h_1 \times m + h_0 \times (m - n)}{n}$$

$$h_3 = \frac{h_1 \times m + h_0 \times (m - x - y) - h_4 \times y}{x}$$

06. 按鍵顏色示例

優先級： **B** > **A**

A：配色方案 -> 配色實例 -> 按鍵文本顏色，當前實例**全局默認**的按鍵文本顏色。

為方便主題聯動設計，建議採用**錨點語法**。

```
# 顏色配置
# 0xAARRGGBB 十六進制顏色代碼 取值範圍 00 - FF
# AA 表示 Alpha 通道，控制透明度，值越低越透明
# RR GG BB 分別表示 紅 綠 藍 通道，值越高飽和度越高
colors:
  11: &color11 0xFF00a0f0 # 藍
  12: &color12 0xFFdd0077 # 玫

# 配色方案
preset_color_schemes:
  # 默認佈局
  default:
  # ...
  key_text_color: *color11 # 按鍵文本顏色，對應錨點 &color11
  kts: *color12 # 特定按鍵文本顏色，對應錨點 &color12
  # ...
```

B：指定某一按鍵採用**局部生效**的按鍵文本顏色。

```
# 鍵盤佈局
preset_keyboards:
  # 默認鍵盤
  default:
  # ...
  keys:
  # ...
  - {click: f, key_text_color: kts}
  # ...
  - {click: j, key_text_color: kts}
  # ...
```

07. 懸浮窗橫排與豎排

以翻月拼音 `luna_pinyin.schema.yaml` 為例，輸入編碼時單次最多可有 **5** 個候選項。

```
# 浮動窗口佈局
layout:
  # 浮動窗口位置：
  # left, right,
  # left_up, right_up,
  # fixed,
  # bottom_left, bottom_right,
  # top_left, top_right
  # ( left, right 需要 Android 5.0 以上 )
  position: right
  movable: once # 是否可移動窗口，或僅移動一次 true | false | once
  all_phrases: false # 所有滿足條件的詞語都顯示在窗口
  min_length: 1 # 候選項進入浮動窗口最短詞長
  max_entries: 5 # 浮動窗口可顯最大詞條數
```

```
sticky_lines: 5 # 固頂行數
max_length: 14 # 候選項單行最大字數，超過字數則換行
min_width: 40 # 浮動窗口最小寬度
max_width: 360 # 浮動窗口最大寬度，超過則自動換行
min_height: 50 # 最小高度
max_height: 400 # 最大高度
```

- 豎排

`position: right` 浮動窗位置可動態跟隨預編輯文本，部分 `miui` 不支持，需將其改為固定 `fixed`

`min_length = 1` 保證所有候選項詞條無論單字或多字，都離開候選欄進入懸浮窗

`max_entries >= 5` 保證所有候選項都留在懸浮窗不會溢出到候選欄

`sticky_lines >= 5` 保證所有候選項單行單條對齊排列

`max_length`、`max_width` 數值儘量大則可避免候選項詞條太長換行展示

- 橫排

`position: right` 浮動窗位置可動態跟隨預編輯文本，部分 `miui` 不支持，需將其改為固定 `fixed`

`min_length = 1` 保證所有候選項詞條無論單字或多字，都離開候選欄進入懸浮窗

`max_entries >= 5` 保證所有候選項都留在懸浮窗不會溢出到候選欄

`sticky_lines = 0` 為零則取消所有候選項的固頂排列

`max_length`、`min_width`、`max_width` 數值儘量大則可避免候選項詞條太長換行展示

08. 同文水平模式

```
# 界面風格
style:
  horizontal: true # 左右方向鍵之水平定位模式，取值 true | false
```

- 水平模式為真，
- 左右鍵可移動編碼、
- 也可移動候選項，
- 上下鍵實現翻頁，
- 翻頁鍵實現翻頁；

- 水平模式為假，
- 左右鍵只移動編碼，
- 上下鍵移動候選項，
- 翻頁鍵實現翻頁。

09. 關閉或隱藏候選欄

① 可以在主題 `xxx.trime.yaml` 中將候選欄高度 `candidate_view_height` 和編碼提示區高度 `comment_height` 設置為零，保存並重啟主題。

```
# 界面風格
style:
  candidate_view_height: 0 # 候選欄高度
  comment_height: 0 # 編碼提示區高度
```

② 也可在主題的鍵盤佈局 `preset_keyboards` 中選一個按鍵綁定事件，如長按或滑動觸發功能鍵 `Candidate_switch`，這個功能鍵可以顯隱候選欄。

```
# 預設按鍵
preset_keys:
  # 顯隱候選欄
  Candidate_switch: {toggle: _hide_candidate, send: Mode_switch, states: [ 有候選, 無候選]}

# 鍵盤佈局
preset_keyboards:
  # 默認鍵盤
  default:
    # ...
    keys:
      # ...
      - {click: keyboard_number, long_click: Menu, swipe_up: Candidate_switch}
    # ...
```

③ 也可在個人方案 `xxx.schema.yaml` 中寫入顯隱候選欄開關 `_hide_candidate`，並將開關狀態默認為隱藏 `reset: 1`。保存後請重新部署，代碼如下：

```
# 方案開關
switches:
  # 顯隱候選欄，，預設： 0 顯示 | 1 隱藏
  - name: _hide_candidate
    reset: 1
    states: [ 有, 無 ]
```

10. 翻頁鍵設定

① 可在配置文件 `default.yaml` 或個人方案 `xxx.schema.yaml` 中設定翻頁鍵。

兩者優先級： 個人方案 > 配置文件

綁定前提條件： 綁定之新按鍵本身無編碼衝突

```
# 按鍵綁定器
key_binder:
  # 綁定配置
  bindings:
    - { when: paging, accept: comma, send: Page_Up } # 逗號上翻
    - { when: has_menu, accept: period, send: Page_Down } # 句號下翻
```

② 可在主題 `xxx.trime.yaml` 的按鍵佈局中選一按鍵綁定 `has_menu` 事件。

```
# 預設按鍵
preset_keys:
  Page_Up: {label: ↑, preview: ↑, repeatable: false, functional: false, send: Page_Up}
  Page_Down: {label: ↓, preview: ↓, repeatable: false, functional: false, send: Page_Down}

# 鍵盤佈局
preset_keyboards:
  # 默認鍵盤
  default:
    # ...
    keys:
      # ...
      - {label: ' ', click: ',', has_menu: Page_Up}
      # ...
      - {label: '.', click: '.', has_menu: Page_Down}
      # ...
```

11. 次選與三選設定

① 可在配置文件 `default.yaml` 或個人方案 `xxx.schema.yaml` 中設定選詞鍵。

兩者優先級：個人方案 > 配置文件

綁定前提條件：綁定之新按鍵本身無編碼衝突

```
# 按鍵綁定器
key_binder:
  # 綁定配置
  bindings:
    - { when: has_menu, accept: comma, send: 2 } # 逗號次選，適合數字鍵不參與編碼的方案
    - { when: has_menu, accept: period, send: 3 } # 句號三選，適合數字鍵不參與編碼的方案

    - { when: has_menu, accept: comma, send: Control+2 } # 逗號次選，適合數字鍵參與編碼的方案
    - { when: has_menu, accept: period, send: Control+3 } # 句號三選，適合數字鍵參與編碼的方案
```

② 可在主題 `xxx.trime.yaml` 的按鍵佈局中選一按鍵綁定 `has_menu` 事件。

```
# 預設按鍵
```



```

preset_keys:

# 適合數字鍵不參與編碼的方案
Second: {label: 次選, preview: 次選, repeatable: false, functional: false,
send: 2}
# 適合數字鍵不參與編碼的方案
Third: {label: 三選, preview: 三選, repeatable: false, functional: false, send:
3}

# 適合數字鍵參與編碼的方案
Second: {label: 次選, preview: 次選, repeatable: false, functional: false,
send: Control+2}
# 適合數字鍵參與編碼的方案
Third: {label: 三選, preview: 三選, repeatable: false, functional: false, send:
Control+3}

# 鍵盤佈局
preset_keyboards:
# 默認鍵盤
default:
# ...
keys:
# ...
- {label: '  ', click: '.', has_menu: Second}
# ...
- {label: ' 。', click: '.', has_menu: Third}
# ...

```

12. 刪詞鍵設定

可在主題 `xxx.trime.yaml` 的按鍵佈局中選一按鍵綁定長按或滑動等事件。

```

# 預設按鍵
preset_keys:
DeleteCandidate: {label: 刪詞, preview: 刪詞, send: Control+Delete}

# 鍵盤佈局
preset_keyboards:
# 默認鍵盤
default:
# ...
keys:
# ...
- {click: Backspace, long_click: DeleteCandidate}
# ...

```

13. 分詞鍵設定

以朙月拼音 `luna_pinyin.schema.yaml` 為例，在輸入狀態將 `shift` 鍵綁定為分詞鍵。

① 打開朙月拼音方案，查找如下段落可知默認的分詞鍵為 `'`，個人方案同理可查。

```
# 拼寫處理器
speller:
  # 上屏隔音符，隔音符號「'」，首位空白用於自動插入至音節邊界處
  delimiter: " '"
```

② 在主題 `xxx.trime.yaml` 的按鍵佈局中為 `shift` 鍵綁定 `has_menu` 事件。

```
# 預設按鍵
preset_keys:
  delimiter: {label: '分詞', preview: '分詞', functional: false, text: ""}

# 鍵盤佈局
preset_keyboards:
  # 默認鍵盤
  default:
    # ...
    keys:
      # ...
      - {click: Shift_L, has_menu: delimiter}
    # ...
```

③ 注意事項

- 個人方案的分詞鍵未必相同，功能鍵 `text` 屬性值保持對應即可，
- 個人主題調用的鍵盤佈局未必為 `default`，也可能是 `qwerty` 等等，需認真核對。

④ 按鍵也可綁定 `composing` 事件

兩者異同：

- `composing`：輸入狀態時，包含 有候選項時 也包含無候選項的 空碼時
- `has_menu`：有候選項時，僅指能呈現 候選項 的狀態，不包含 空碼時

14. 迴車輸出注音符號

① 可在注音方案 `bopomofo.schema.yaml` 中設定迴車鍵直接輸出注音符號。

按需選用如下任意一條綁定指令，前兩者較為常用。

```
# 按鍵綁定器
key_binder:
  # 綁定配置
  bindings:
    # 輸出上屏原始碼，語句流方案中與下者不同，非語句流方案等同
    - { when: has_menu, accept: Return, send: Control+Return }
    # 輸出上屏腳本碼，語句流方案中與上者不同，非語句流方案等同
    - { when: has_menu, accept: Return, send: Shift+Return }
    # 輸出提示碼
    - { when: has_menu, accept: Return, send: Control+Shift+Return }
```

② 可在主題 `xxx.trime.yaml` 的按鍵佈局中選一按鍵綁定 `has_menu` 或 `long_click` 等事件。

```
# 預設按鍵
preset_keys:
  # 原始碼
  CommitRawInput: {label: 編碼, preview: 編碼, send: Control+Return}
  # 腳本碼
  CommitScriptText: {label: 編碼, preview: 編碼, send: Shift+Return}
  # 提示碼
  CommitComment: {label: 編碼, preview: 編碼, send: Control+Shift+Return}

# 鍵盤佈局
preset_keyboards:
  # 默認鍵盤
  default:
    # ...
    keys:
      # ...
      - {click: Return, has_menu: CommitScriptText}
```

15. 指定鍵盤佈局

以朙月拼音 `luna_pinyin.schema.yaml` 和預置主題 `trime.yaml` 為例

① 指定鍵盤佈局

I 導入指定鍵盤佈局

直接引用指定鍵盤，部署後主題文件不生成全新鍵盤佈局的代碼塊，

極端場景下可能引用失效，此時將智能調用 `.default` 鍵盤佈局。

```
# 鍵盤佈局
preset_keyboards:
  # ...
  # 預設 36 鍵
  qwerty0:
    # ...
    # 朙月拼音
    luna_pinyin:
      # 導入預設佈局
      import_preset: qwerty0
```

II 嵌入指定鍵盤佈局

嵌入指定鍵盤的代碼塊，部署後主題文件將生成全新鍵盤佈局的代碼塊。

筆者個人推薦這種語法操作。

```
# 鍵盤佈局
preset_keyboards:
  # ...
  # 預設 36 鍵
  qwerty0:
    # ...
    # 朙月拼音
    luna_pinyin:
      # 嵌入指定代碼塊
      __include: /preset_keyboards/qwerty0
```

② 註冊朙月拼音鍵盤，此步操作非必須但建議。

```
# 界面風格
style:
  # 鍵盤配置
  keyboards: [.default, luna_pinyin, letter, number, symbols]
```

16. 設計衍生鍵盤

以同文風主題 `tongwenfeng.trime.yaml` 和倉頡五代方案 `cangjie5.schema.yaml` 為例，設計一款倉頡衍生鍵盤。

① 打開主題，找到 `preset_keyboards/default` 段落，此鍵盤佈局為主題默認，本次將以此為模板設計衍生佈局。

```
# 键盘布局
preset_keyboards:
  # 默認鍵盤
  default:
    author: "暖暖"
    name: 26键默认布局
    width: 10
    # height: 48
    ascii_mode: 0
    keys:
      # 第一行
      - {click: q, long_click: 1, key_back_color: bh1, key_text_color: th1}
      - {click: w, long_click: 2, key_back_color: bh1, key_text_color: th1}
      - {click: e, long_click: 3, key_back_color: bh1, key_text_color: th1}
      - {click: r, long_click: 4, key_back_color: bh1, key_text_color: th1}
      - {click: t, long_click: 5, key_back_color: bh1, key_text_color: th1}
      - {click: y, long_click: 6, key_back_color: bh1, key_text_color: th1}
      - {click: u, long_click: 7, key_back_color: bh1, key_text_color: th1}
      - {click: i, long_click: 8, key_back_color: bh1, key_text_color: th1}
      - {click: o, long_click: 9, key_back_color: bh1, key_text_color: th1}
      - {click: p, long_click: 0, key_back_color: bh1, key_text_color: th1}

      # 第二行
      - {width: 5, key_back_color: bh2, key_text_color: th2}
      - {click: a, long_click: select_all, key_back_color: bh2, key_text_color: th2}

th2}
      - {click: s, long_click: '@', key_back_color: bh2, key_text_color: th2}
      - {click: d, long_click: '#', key_back_color: bh2, key_text_color: th2}
```

```

- {click: f, long_click: '$', key_back_color: bh2, key_text_color: th2}
- {click: g, long_click: '%', swipe_down: "date_time", key_back_color:
bh2, key_text_color: th2}
- {click: h, long_click: '!', swipe_down: "+", key_back_color: bh2,
key_text_color: th2}
- {click: j, long_click: '&', swipe_down: "-", key_back_color: bh2,
key_text_color: th2}
- {click: k, long_click: '*', swipe_down: "_", key_back_color: bh2,
key_text_color: th2}
- {click: l, long_click: '(){Left}', swipe_down: "=", key_back_color: bh2,
key_text_color: th2}
- {width: 5, key_back_color: bh2, key_text_color: th2}

# 第三行
- {click: Shift_L, composing: delimiter, width: 15,
hilited_key_back_color: benter, key_back_color: bbs, key_text_color: tbs}
- {click: z, long_click: '`', swipe_down: '[]{Left}', key_back_color: bh3,
key_text_color: th3}
- {click: x, long_click: cut, swipe_down: '{}{Left}', key_back_color: bh3,
key_text_color: th3}
- {click: c, long_click: copy, swipe_down: ' ', key_back_color: bh3,
key_text_color: th3}
- {click: v, long_click: paste, swipe_down: "\\ ", key_back_color: bh3,
key_text_color: th3}
- {click: b, long_click: ";", key_back_color: bh3, key_text_color: th3}
- {click: n, long_click: ":", key_back_color: bh3, key_text_color: th3}
- {click: m, long_click: '?', swipe_down: "|", key_back_color: bh3,
key_text_color: th3}
- {click: Backspace, width: 15, key_back_color: bbs, key_text_color: tbs}

# 第四行
- {click: Keyboard_number, long_click: Menu, width: 12.5, key_back_color:
bgn, key_text_color: tgn}
- {click: Keyboard_bqrw, long_click: Theme_settings, width: 12.5,
key_text_size: "18", key_back_color: bgn, key_text_color: tgn}
- {click: ',', label: '·', long_click: '<>{Left}', key_back_color: bh4,
key_text_color: th4}
- {click: space, long_click: Mode_switch, swipe_left: "Left", swipe_right:
"Right", swipe_up: Schema_switchcn, width: 30, key_back_color: bkg,
key_text_color: tkg}
- {click: '.', label: '。', long_click: Keyboard_kao, key_back_color: bh4,
key_text_color: th4}
- {click: Henkan, long_click: '?', key_text_size: "18", key_back_color:
bh4, key_text_color: th4}
- {click: Return, swipe_up: Escape, width: 15, key_back_color: benter,
key_text_color: tenter}

```

② 同層級下，新建倉頡五代同名鍵盤佈局 `cangjie5`，善用 `__include` 與 `__patch` 語法，設計衍生鍵盤。

從第一排第一顆按鍵 `@0` 開始排序，逐顆按鍵定義按鍵標籤 `label` 與按鍵助記 `hint` 屬性。

每顆按鍵無論寬度多少均算一個元素，佈局空白 `- {width: 5}` 也算一個元素。

```

# 倉頡五代
cangjie5:
author: 聽雨客

```

```
name: 倉頡五代
label_transform: none
__include: /preset_keyboards/default
__patch:
  # 第一行
  keys/@0/label: '手' # q
  keys/@0/hint: 'q' # q
  keys/@1/label: '田' # w
  keys/@1/hint: 'w' # w
  keys/@2/label: '水' # e
  keys/@2/hint: 'e' # e
  keys/@3/label: '口' # r
  keys/@3/hint: 'r' # r
  keys/@4/label: '廿' # t
  keys/@4/hint: 't' # t
  keys/@5/label: '卜' # y
  keys/@5/hint: 'y' # y
  keys/@6/label: '山' # u
  keys/@6/hint: 'u' # u
  keys/@7/label: '戈' # i
  keys/@7/hint: 'i' # i
  keys/@8/label: '人' # o
  keys/@8/hint: 'o' # o
  keys/@9/label: '心' # p
  keys/@9/hint: 'p' # p

  # 第二行
  # keys/@10/label: '' # 空白
  keys/@11/label: '日' # a
  keys/@11/hint: 'a' # a
  keys/@12/label: '尸' # s
  keys/@12/hint: 's' # s
  keys/@13/label: '木' # d
  keys/@13/hint: 'd' # d
  keys/@14/label: '火' # f
  keys/@14/hint: 'f' # f
  keys/@15/label: '土' # g
  keys/@15/hint: 'g' # g
  keys/@16/label: '竹' # h
  keys/@16/hint: 'h' # h
  keys/@17/label: '十' # j
  keys/@17/hint: 'j' # j
  keys/@18/label: '大' # k
  keys/@18/hint: 'k' # k
  keys/@19/label: '中' # l
  keys/@19/hint: 'l' # l
  # keys/@20/label: '' # 空白

  # 第三行
  # keys/@21/label: '檔' # Shift_L
  keys/@22/label: '符' # z
  keys/@22/hint: 'z' # z
  keys/@23/label: '難' # x
  keys/@23/hint: 'x' # x
  keys/@24/label: '金' # c
  keys/@24/hint: 'c' # c
  keys/@25/label: '女' # v
  keys/@25/hint: 'v' # v
```

```

keys/@26/label: '月' # b
keys/@26/hint: 'b' # b
keys/@27/label: '弓' # n
keys/@27/hint: 'n' # n
keys/@28/label: '一' # m
keys/@28/hint: 'm' # m
# keys/@29/label: '退' # BackSpace

# 第四行
# keys/@30/label: '數' # Keyboard_number
keys/@30/has_menu: Page_Up
# keys/@31/label: '' # Keyboard_bqrw
keys/@31/has_menu: Page_Down
# keys/@32/label: '逗' # ,
# keys/@33/label: '' # space
# keys/@34/label: '句' # .
# keys/@35/label: '' # space_R2
# keys/@36/label: '迴' # Return

```

③ 註冊倉頡五代佈局，名稱同於上一步操作中節點名 `cangjie5`，此步操作非必須但建議。

```

# 界面風格、功能開關
style:
# 鍵盤配置
keyboards: [.default, letter, default, cangjie5, number, symbols, edit, move,
bqrw1, bqrw2, bqrw3, bqrw4, bqrw5, bqqt1, bqqt2, bqqt3, bqqt4, bqz1, bqz2,
bqz3, bqz4, bqz5, bqpt1, bqpt2, bqhd1, bqhd2, bqhc1, bqhc2, bqlx1, bqlx2,
bqlx3, bqlx4, bqlx5, bqkj1, bqkj2, bqkj3, bqkj4, bqfh1, bqfh2, bqfh3, bqfh4,
kao_fond, kao_sad, kao_mad, kao_astonished, kao_impotent, kao_happy, numberb]

```

17. 快捷符號啟用

① 以同文原版為例，確認 `/storage/emulated/0/rime` 路徑下包含 `symbols.yaml` 文件；

② 以朙月拼音 `luna_pinyin.schema.yaml` 為例，分段落寫入如下代碼至個人方案

`xxx.schema.yaml`：

```

# 句讀處理器
punctuator:
# 導入外部文件
import_preset: symbols

# 識別器
recognizer:
# 樣式匹配
patterns:
# 快捷符號
punct: "^/([0-9]0?|[A-Za-z]+)$"

```

③ 保存文件，返回同文 `app` 點擊部署。

18. 反查依賴之反查翻譯器

以翻月拼音方案 `luna_pinyin.schema.yaml` 和同文原版為例，此方案採用五筆畫 `stroke` 作為反查。

① 確認 `/storage/emulated/0/rime` 路徑下包含：

`stroke.schema.yaml`

`stroke.dict.yaml`

② 分段落寫入如下代碼至個人方案 `xxx.schema.yaml`：

```
# 方案信息
schema:
  # 備用方案依賴
  dependencies:
    - stroke

# 輸入引擎
engine:
  # 翻譯器
  translators:
    # 反查翻譯器
    - reverse_lookup_translator

# 反查配置
reverse_lookup:
  # 反查詞典設定
  dictionary: stroke
  # 提前顯示尚未輸入完整碼的字〔僅 table_translator 有效〕
  enable_completion: true
  # 前綴
  prefix: ""
  # 後綴
  suffix: ""
  # 開始提示符
  tips: [筆畫]
  # 上屏碼格式
  preedit_format:
    - xlit/hspnz/一丨丿乙/
  # 提示碼格式
  comment_format:
    - xform/([nl])v/$1ü/

# 識別器
recognizer:
  # 樣式匹配
  patterns:
    # 反查引導
    reverse_lookup: "`[a-z]*'?$"
```

③ 保存文件，返回同文 `app` 點擊部署。

④ 臨時獲取不到五筆畫相關文件，或暫不需反查功能，可先將相關代碼用 `#` 註釋後再重新部署。

```
# 方案信息
schema:
```



```

# 備用方案依賴
# dependencies:
# - stroke

# 輸入引擎
engine:
# 翻譯器
translators:
# 反查翻譯器
# - reverse_lookup_translator

# 反查配置
# reverse_lookup:
# 反查詞典設定
# dictionary: stroke
# 提前顯示尚未輸入完整碼的字〔僅 table_translator 有效〕
# enable_completion: true
# 前綴
# prefix: ""
# 後綴
# suffix: ""
# 開始提示符
# tips: 〔筆畫〕
# 上屏碼格式
# preedit_format:
# - xlit/hspnz/一丨丿乙/
# 提示碼格式
# comment_format:
# - xform/([nl])v/$1ü/

# 識別器
recognizer:
# 樣式匹配
patterns:
# 反查引導
# reverse_lookup: "[a-z]*'?$"

```

19. 反查依賴之反查過濾器

反查翻譯器的演進版本，個人方案 `xxx.schema.yaml` 可掛載多個反查實例。

① 確認 `/storage/emulated/0/rime` 路徑下包含：

```
luna_pinyin.schema.yaml
```

```
luna_pinyin.dict.yaml
```

```
cangjie5.schema.yaml
```

```
cangjie5.dict.yaml
```

② 打開個人方案依次寫入如下代碼：

```

# 1. 方案信息
schema:
# 備用方案依賴
dependencies:

```

- luna_pinyin
- cangjie5

2. 輸入引擎

engine:

分段器

segmentors:

- affix_segmentor@luna_pinyin # 詞綴分段器甲，標識 luna_pinyin
- affix_segmentor@cangjie5 # 詞綴分段器乙，標識 cangjie5

翻譯器

translators:

- # 原有「script_translator」或「table_translator」主翻譯器保留
- # 追加如下副翻譯器
- script_translator@luna_pinyin # 副腳本翻譯器，標識 luna_pinyin
- table_translator@cangjie5 # 副表格翻譯器，標識 cangjie5

過濾器

filters:

- reverse_lookup_filter@reverse_lookup # 反查過濾器

3-1. 副腳本翻譯器

對應 script_translator@luna_pinyin

luna_pinyin:

- # 作用標識，對應 affix_segmentor@luna_pinyin
- tag: luna_pinyin
- # 翻譯詞典設定
- dictionary: luna_pinyin
- # 前綴
- prefix: "`"
- # 後綴
- suffix: ";"
- # 開始提示符
- tips: [拼音]
- # 結束提示符
- closing_tips: [同文]
- # 上屏碼格式
- preedit_format:
 - xform/([nl])v/\$1ü/
 - xform/([nl])ue/\$1üe/
 - xform/([jqxy])v/\$1u/

3-2. 副表格翻譯器

對應 table_translator@cangjie5

cangjie5:

- # 作用標識，對應 affix_segmentor@cangjie5
- tag: cangjie5
- # 翻譯詞典設定
- dictionary: cangjie5
- # 是否啟用用戶詞典
- enable_user_dict: false
- # 前綴
- prefix: "~"
- # 後綴
- suffix: ";"
- # 開始提示符
- tips: [倉頡]
- # 結束提示符
- closing_tips: [同文]
- # 上屏碼格式

```

preedit_format:
- 'xform/^[a-z]*)$/$1\t (\U$1\E) /'
- "xlit|ABCDEFGHIJKLMNopqrstuvwxyz|日月金木水火土竹戈十大中一弓人心手口尸廿山女田難
卜符|"
# 提示碼格式
comment_format:
- "xlit|abcdefghijklmnopqrstuvwxyz~|日月金木水火土竹戈十大中一弓人心手口尸廿山女田
難卜符~|"
# 禁用指定樣式編碼存入用戶詞典
disable_user_dict_for_patterns:
- "^z.*$"
- "^yyy.*$"

# 4. 反查配置
# 對應 reverse_lookup_filter@reverse_lookup
reverse_lookup:
# 作用標識範圍
# 對應 affix_segmentor@luna_pinyin 與 affix_segmentor@cangjie5
tags: [ luna_pinyin, cangjie5 ]
# 反查詞典設定，對應個人方案主翻譯器之詞典名「translator/dictionary」
dictionary: xxx
# 重寫提示碼，是否覆蓋其他提示，可重寫主翻譯器提示碼「translator/comment_format」
overwrite_comment: false
# 提示碼格式
comment_format:
# TODO 格式視情況而定，可參考主翻譯器之提示碼「translator/comment_format」
# ...

# 5. 識別器
recognizer:
# 樣式匹配
patterns:
# 鬮月反查
luna_pinyin: "`[a-z]*;? $"
# 倉頡反查
cangjie5: "~[a-z]*;? $"

```

③ 保存文件，返回同文 `app` 點擊部署。

20. 參考

(1) [鬮月拼音方案](#)

(2) [粵語拼音方案](#)

(3) [Rime 方案詳解](#)

(4) [Trime 主題詳解](#)