This repository Search Pull requests Issues Gist

tmk / **tmk_keyboard** Watch 113 Star 823 Fork 551

FAQ Keymap

Edit New Page

tmk edited this page 4 days ago · 32 revisions

READ FIRST

https://github.com/tmk/tmk_keyboard/blob/master/doc/keymap.md

How to get keycode

See doc/keycode.txt. Keycodes are actually defined in common/keycode.h.

Sysrq key

Use keycode for Print Screen(`KC_PSCREEN` or `KC_PSCR`) instead of `KC_SYSREQ` . Key combination of 'Alt + Print Screen' is recognized as 'System request'.

See issue #168 and

- http://en.wikipedia.org/wiki/Magic_SysRq_key
- http://en.wikipedia.org/wiki/System_request

Power key doesn't work

Use `KC_PWR` instead of `KC_POWER` or vice versa.

- `KC_PWR` works with Windows and Linux, not with OSX.
- `KC_POWER` works with OSX and Linux, not with Windows.

http://geekhack.org/index.php?topic=14290.msg1327264#msg1327264

Oneshot modifier

Solves my personal 'the' problem. I often got 'the' or 'THe' wrongly instead of 'The'. Oneshot Shift mitgates this for me. https://github.com/tmk/tmk_keyboard/issues/67

Modifier/Layer stuck

Modifier keys or layers can be stuck unless layer switching is configured properly. For Modifier keys and layer actions you have to place `KC_TRANS` on same position of destination layer to unregister the modifier key or return to previous layer on release event.

- https://github.com/tmk/tmk_keyboard/blob/master/doc/keymap.md#31-momentary-switching
- http://geekhack.org/index.php?topic=57008.msg1492604#msg1492604

Pages 10

- Home
- Compatible Microcontrollers
- FAQ
- FAQ Build
- FAQ Keymap
- Git subtree
- HHKB Alt controller
- mbed cortex porting
- Previously Asked Questions
- Projects

Clone this wiki locally

https://github.com/tmk/tn

Clone in Desktop

- https://github.com/tmk/tmk_keyboard/issues/248

Mechanical Lock Switch Support

https://github.com/tmk/tmk_keyboard#mechanical-locking-support

This feature is for *mechanical lock switch* like this Alps one.
http://deskthority.net/wiki/Alps_SKCL_Lock

Using enabling this feature and using keycodes `LCAP`, `LNUM` or `LSCR` in keymap you can use physical locking CapsLock, NumLock or ScrollLock keys as you expected.

Old vintage mechanical keyboards occasionally have lock switches but modern ones don't have. ***You don't need this feature in most case and just use keycodes `CAPS`, `NLCK` and `SLCK`.***

Input special charactors other than ASCII like Cédille 'Ç'

NO UNIVERSAL METHOD TO INPUT THOSE WORKS OVER ALL SYSTEMS. You have to define **MACRO** in way specific to your OS or layout.

See this post for example **MACRO** code.

http://deskthority.net/workshop-f7/tmk-keyboard-firmware-collection-t4478-120.html#p195620

On **Windows** you can use `AltGr` key or **Alt code**.

- http://en.wikipedia.org/wiki/AltGr_key
- http://en.wikipedia.org/wiki/Alt_code

On **Mac** OS defines `Option` key combinations.

- http://en.wikipedia.org/wiki/Option_key#Alternative_keyboard_input

On **Xorg** you can use `compose` key, instead.

- http://en.wikipedia.org/wiki/Compose_key

And see this for **Unicode** input.

- http://en.wikipedia.org/wiki/Unicode_input

Apple keyboard Fn

Not supported.

Apple keyboard sends keycode for Fn unlike most of other keyboards. I think you can send Apple Fn key using Apple venter specific Page 0xff00 and usage 0x0003. But you have to change HID Report Descriptor for this, of course.

https://opensource.apple.com/source/IOHIDFamily/IOHIDFamily-606.1.7/IOHIDFamily/AppleHIDUsageTables.h

Media control keys in Mac OSX

KC_MNXT and KC_MPRV does not work on Mac

Use `KC_MFFD` (`KC_MEDIA_FAST_FORWARD`) and `KC_MRWD` (`KC_MEDIA_REWIND`) instead of `KC_MNXT` and `KC_MPRV` . See https://github.com/tmk/tmk_keyboard/issues/195

Keys supported in Mac OSX?

You can know which keycodes are supported in OSX from this source code.

`usb_2_adb_keymap` array maps Keyboard/Keypad Page usages to ADB scancodes(OSX internal keycodes).

https://opensource.apple.com/source/IOHIDFamily/IOHIDFamily-606.1.7/IOHIDFamily/Cosmo_USB2ADB.c

And `IOHIDConsumer::dispatchConsumerEvent` handles Consumer page usages.

https://opensource.apple.com/source/IOHIDFamily/IOHIDFamily-606.1.7/IOHIDFamily/IOHIDConsumer.cpp

JIS keys in Mac OSX

Japanese JIS keyboard specific keys like `無変換(Muhenkan)` , `変換(Henkan)` , `ひらがな(hiragana)` are not recognized on OSX. You can use **Seil** to enable those keys, try following options.

- Enable NFER Key on PC keyboard
- Enable XFER Key on PC keyboard
- Enable KATAKANA Key on PC keyboard

https://pqrs.org/osx/karabiner/seil.html

RN-42 Bluetooth doesn't work with Karabiner

Karabiner - Keymapping tool on Mac OSX - ignores inputs from RN-42 module by default. You have to enable this option to make Karabiner working with your keyboard. https://github.com/tekezo/Karabiner/issues/403#issuecomment-102559237

See these for the deail of this problem. https://github.com/tmk/tmk_keyboard/issues/213 https://github.com/tekezo/Karabiner/issues/403

Esc and `~ on a key

You can define FC660 and Poker style ESC with `ACTION_LAYER_MODS` . https://github.com/tmk/tmk_keyboard/blob/master/doc/keymap.md#35-momentary-switching-with-modifiers

```
#include "keymap_common.h"


/* Leopold FC660
```

```
 * https://elitekeyboards.com/products.php?sub=leopold,compact&pid=fc660c
 * Shift + Esc = ~
 * Fn    + Esc = `
 *
 * Votex Poker II
 * https://adprice.fedorapeople.org/poker2_manual.pdf
 * Fn         + Esc = `
 * Fn + Shift + Esc = ~
 */
const uint8_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {
    /* 0: qwerty */
    [0] = KEYMAP( \
        ESC, 1,   2,   3,   4,   5,   6,   7,   8,   9,   0,   MINS,EQL, N
        TAB, Q,   W,   E,   R,   T,   Y,   U,   I,   O,   P,   LBRC,RBRC,B
        LCTL,A,   S,   D,   F,   G,   H,   J,   K,   L,   SCLN,QUOT,ENT,
        FN0, NUBS,Z,   X,   C,   V,   B,   N,   M,   COMM,DOT, SLSH,RSFT,E
        LCTL,LGUI,LALT,          SPC,                     RALT,FN1, RGUI,R
    [1] = KEYMAP( \
        GRV, TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,T
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,T
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,T
        TRNS,TRNS,TRNS,          TRNS,                    TRNS,TRNS,TRNS,T
    [2] = KEYMAP( \
        GRV, F1,  F2,  F3,  F4,  F5,  F6,  F7,  F8,  F9,  F10, F11, F12, T
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,T
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,T
        TRNS,TRNS,TRNS,          TRNS,                    TRNS,TRNS,TRNS,T
};

const uint16_t PROGMEM fn_actions[] = {
    // https://github.com/tmk/tmk_keyboard/blob/master/doc/keymap.md#35-mo
    [0] = ACTION_LAYER_MODS(1, MOD_LSFT),
    [1] = ACTION_LAYER_MOMENTARY(2),
};
```

Otherwise, you can write code, see this.
https://github.com/p3lim/keyboard_firmware/commit/fd799c12b69a5ab5addd1d4c03380a1b8ef8e9dc

32 Fn keys are not enough?

actionmap

It uses 16 bit codes and has no limitation of 32 Fn at the expense of memory space. TMK keymap is actually is 8 bit codes as subset of the actionmap.
https://github.com/tmk/tmk_keyboard/issues?utf8=%E2%9C%93&q=is%3Aissue+actionmap

extension for modified keys

https://geekhack.org/index.php?topic=41989.msg1885526#msg1885526

Arrow on Right Modifier keys with Dual-Role

This turns right modifer keys into arrow keys when the keys are tapped while still modifiers when the keys are hold. In TMK the dual-role function is dubbed **TAP**.

```
#include "keymap_common.h"


/* Arrow keys on right modifier keys with TMK dual role feature
 *
 *  https://github.com/tmk/tmk_keyboard/blob/master/doc/keymap.md#213-modi
 *  https://en.wikipedia.org/wiki/Modifier_key#Dual-role_keys
 */
const uint8_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {
    /* 0: qwerty */
    [0] = KEYMAP( \
        ESC, 1,   2,   3,   4,   5,   6,   7,   8,   9,   0,   MINS,EQL, N
        TAB, Q,   W,   E,   R,   T,   Y,   U,   I,   O,   P,   LBRC,RBRC,B
        LCTL,A,   S,   D,   F,   G,   H,   J,   K,   L,   SCLN,QUOT,ENT,
        LSFT,NUBS,Z,   X,   C,   V,   B,   N,   M,   COMM,DOT, SLSH,FN0, E
        FN4, LGUI,LALT,          SPC,                     APP, FN2, FN1, F
    [1] = KEYMAP( \
        GRV, F1,  F2,  F3,  F4,  F5,  F6,  F7,  F8,  F9,  F10, F11, F12, T
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,T
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,
        TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,TRNS,FN5, T
        TRNS,TRNS,TRNS,          TRNS,                    TRNS,FN7, FN6, F
};

const uint16_t PROGMEM fn_actions[] = {
    [0] = ACTION_MODS_TAP_KEY(MOD_RSFT, KC_UP),
    [1] = ACTION_MODS_TAP_KEY(MOD_RGUI, KC_DOWN),
    [2] = ACTION_MODS_TAP_KEY(MOD_RALT, KC_LEFT),
    [3] = ACTION_MODS_TAP_KEY(MOD_RCTL, KC_RIGHT),
    [4] = ACTION_LAYER_MOMENTARY(1),
    [5] = ACTION_MODS_TAP_KEY(MOD_RSFT, KC_PGUP),
    [6] = ACTION_MODS_TAP_KEY(MOD_RGUI, KC_PGDN),
    [7] = ACTION_MODS_TAP_KEY(MOD_RALT, KC_HOME),
    [8] = ACTION_MODS_TAP_KEY(MOD_RCTL, KC_END),
};
```

Dual-role key: https://en.wikipedia.org/wiki/Modifier_key#Dual-role_keys