参考

https://kiyooka.hatenablog.com/entry/2018/11/18/042524
https://qiita.com/sirojake/items/2e1c45f346b9ba3b9e7f
https://qiita.com/chesscommands/items/42ba11edd0f9b183ead7
https://note.mu/huaa/n/nd26474ae5b08
https://leico.github.io/TechnicalNote/QMK/key-customize
https://arekore.netlify.com/2018 /0204 -%E3 %82 %AD%E3 %83 %BC%E3 %83 %9 C%E3 %83 %BC%E3 %83%89%E4%BD%9C%E3%82%8A%E3%81%BE%E3%81%97%E3%81%9F/
https://scrapbox.io/self-made-kbds-ja/XD75
https://temcee.hatenablog.com/entry/diy_key_board
https://yhara.jp/LetsSplitKeySwitches
https://input.club/the-comparative-guide-to-mechanical-switches/
https://ysa256.blogspot.com/2018/06/xd752.html
https://scrapbox.io/self-made-kbds-ja/XD75

■ファームウェアビルド

https://ascii.jp/elem/000/001/645/1645504/
http://okapies.hateblo.jp/entry/2019/02/02/133953
https://beta.docs.qmk.fm/features/feature_macros
https://qiita.com/Nymphium/items/2775b8c3555f733fafdc

■日本語キーボードにする場合

https://note.mu/huaa/n/nd26474ae5b08

必要なもの

パーツ	説明
PCB	ダイオードやマイコンを半田付けする基盤
ダイオード	電流の流れを制御
マイコン	ファームウェアをインストールし、キーボードとして動作させる。
プレート	スイッチを固定するもの
スイッチ	キーのスイッチ部。赤軸とか茶軸とかでキーの重さや音が変わる。
キーキャップ	文字が書いてあるスイッチに被せるもの
ケース	キーボード自体のケース

PCB、ダイオード、マイコンセットになっているキットも存在する。
半田付け不要で、組み合わせるだけで良いものものある。

XD75re

概要

5x15 の 75 キーで 60% キーボードサイズのキーボード。
基盤にはダイオードや LED が実装済みで販売されている。
キースイッチはホットスワップで半田付け不要なので、扱いやすい。
好きなキーキャップとキースイッチがあれば、簡単に組み立てが完了する。

種類	キースイッチ	半田付け
XD75Re	MX のみ	半田付け不要で、キースイッチはホットスワップ
XD75Am	MX、ALPS 両対応	半田付けが必要

注文

注文したもの	AliExpress の URL	備考
XD75RE の基盤	https://ja.aliexpress.com/item/32818745981.html?spm=a2g0s.9042311.0.0.17b94c4d	XD75Re を選択 (黒い基盤)、白い方は XD75Am で半田付けが必要になるので注意
ケース	https://ja.aliexpress.com/item/32821659665.html?spm=a2g0s.9042311.0.0.17b94c4d	アクリルボードも付いてくる。
プレート	https://ja.aliexpress.com/item/32821198488.html?spm=a2g0s.9042311.0.0.17b94c4d	上記のケースがあれば、このプレートは不要。無駄になってしまった。
スイッチ	https://ja.aliexpress.com/item/32815810102.html?spm=a2g0s.9042311.0.0.17b94c4d	茶軸を 80 個ほど購入。静音茶軸の方が良かったかも。
キーキャップ	https://ja.aliexpress.com/item/32997912293.html?spm=a2g0s.9042311.0.0.17b94c4d	無難。さわり心地も悪くない。ただ、ctrl とか alt キーが少ない

組み立て

基本的に説明書はない。

1. ケースの裏に基盤を当てて、四隅だけにキースイッチを取り付けて基盤を仮止めする
    1. ※ケースにスイッチを全てはめてから、基盤を付けるとやりにくいので注意
2. キースイッチをひとつづつ取り付け
3. ネジとナットで基盤を固定。ネジが短いのでアクリルボードはそのままナットを締めることはできない。
4. 両サイドのネジ穴で基盤とアクリルボードを固定

最終的にケースは使わず、プレートのみにした。
ケースを使うと高さが高くなり、ちょっとゴツくなる。プレートのみだとコンパクトでスタイリッシュに見えるため。

■ケースを使った場合

■ケースを使わず、プレートのみの場合

プログラム

qmk firmware でファームウェアの書き換えが可能。
https://github.com/qmk/qmk_firmware

■コンパイル、ファームウェアインストールの流れ (docker を使う場合)

git からクローン

```
git clone https://github.com/qmk/qmk_firmware.git
```

```
qmk_firmware/keyboards/xd75/keymaps
```

以下に幾つかサンプル的なプログラムがある。

```
qmk_firmware/keyboards/xd75/keymaps/default
```

が標準のキー配置。この default をインストールする場合は

1. ./util/docker_build.sh xd75:default:dfu
2. キーボード側をリセットする
    1. リセットする方法１：基盤の真ん中のスイッチをショートさせる。ケースの真ん中に穴が開いている。ここを除くと２つの穴が見えるので、細いクリップ等でショートさせる。
    2. (基盤を裏から見ると真ん中に 6 つのスイッチが縦に並んでいる。この一番上の２つ。表からショートさせるのが難しい場合は裏からショートさせると楽)
    3. リセットする方法２：キー配置を変更する際に、リセットキーを配置することが出来る。このキーで物理的にショートしなくてもリセットが出来る。

■コンパイル、ファームウェアインストールの流れ (docker 以外の場合)

git からクローン

```
git clone https://github.com/qmk/qmk_firmware.git
```

コンパイルに必要なパッケージを準備する。
パッケージをインストールするコマンドが準備されている。環境によりコマンドが違う。

```
./util/linux_install.sh
```

とか

```
./util/mac_install.sh
```

とか
パッケージが準備されたら

```
make xd75:default:dfu
```

キーボードのリセット等は上記の docker を利用する場合と同じ。

■自分独自のキー配置やマクロを作成する

もし自分用のキー配置やマクロを作成する場合は適当なディレクトリを

```
qmk_firmware/keyboards/xd75/keymaps/hoge
```

などを作成して

```
config.h
keymap.c
readme.md
rules.mk
```

を default からコピーする。keymap.c を書き換えて

```
./util/docker_build.sh xd75:hoge:dfu
make  xd75:hoge:dfu
```

qmk_firmware の ドキュメント

https://docs.qmk.fm/

https://github.com/qmk/qmk_firmware/tree/master/docs

qmk_firmware で同時押しを実現する

qmk firmware ではキー押してレイアウトを変更しながら、順番で好きな動作をさせることが出来るが、順番に関係なく同時押しで動きを変えたいことがある。
やり方が正しいかどうかは分からないが、同時押しで動きを変えるには

1. 物理的に押されている／離されているキーを記憶
2. 物理的に押されている／離されているキーから判断して、論理的に押したことにしたい／離されたことにしたいキーの信号を送信

をすれば同時押しを実現可能。

■サンプル

Linux、Windows の場合は、Ctrl + 矢印キー で単語移動、段落移動、Ctrl + Shift + 矢印キー で単語選択、段落選択が標準的な動きだけど
Mac の場合は、Option キーだったり、Cmd キーだったりでわかりにくいので、Cmd + 矢印キーで単語移動や Home、End で行末移動を出来るようにする。
具体的には、Mac 用のレイヤーにしているときは、物理的に Cmd と矢印キーが押された場合に、他のキーに置き換えてしまう。

```
/* Copyright 2017 Wunder
*
* This program is free software: you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation, either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/
#include QMK_KEYBOARD_H

#define JP_ZHTG KC_GRV  // hankaku/zenkaku|kanzi
#define JP_YEN  KC_INT3 // yen, |
```

```
#define JP_CIRC KC_EQL  // ^, ~
#define JP_AT   KC_LBRC // @, `
#define JP_LBRC KC_RBRC // [, {
#define JP_COLN KC_QUOT // :, *
#define JP_RBRC KC_NUHS // ], }
#define JP_BSLS KC_INT1 // \, _

// Layer shorthand
#define _USB 0
#define _JPB 1

// Layer shorthand
#define _US1 2
#define _JP1 3
#define _JP2 4

// Layer For Mac
#define _MUB 5
#define _MJB 6
#define _MU1 7
#define _MJ1 8
#define _MJ2 9
#define _MJ3 10

// Layer For Common
#define __CS 11
#define __FN 12
#define _FN2 13
#define _FN3 14
// #define _MCS 4
// #define _MFN 5
#define _MNT 15
#define _TMP 16

#define KC_WMH LT(__CS, KC_MHEN)
#define KC_WHN LT(__FN, KC_HENK)
#define KC_MMH LT(__CS, KC_LANG2)
#define KC_MHN LT(__FN, KC_LANG1)

#define KEYS_MAX 256
#define PHYS 0
#define LOGI 1
#define TRAN 2
bool keys[KEYS_MAX][3] = {};

#define MODE_NORMAL 0
#define MODE_MAC 1
uint8_t mode = MODE_NORMAL;
uint8_t layer = _USB;

// Defines the keycodes used by our macros in process_record_user
// とりあえず、使わない
enum custom_keycodes {
  QMKBEST = SAFE_RANGE,
  QMKURL
};


const uint16_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {
  /* QWERTY(US)
                                                                                                                                                *
.-----------------------------------------------------------------------------------------------------------------------------------------------
```

■サンプル 2

```
/* Copyright 2017 Wunder
*
* This program is free software: you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation, either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

```
*/
#include QMK_KEYBOARD_H

#define JP_ZHTG KC_GRV  // hankaku/zenkaku|kanzi
#define JP_YEN  KC_INT3 // yen, |
#define JP_CIRC KC_EQL  // ^, ~
#define JP_AT   KC_LBRC // @, `
#define JP_LBRC KC_RBRC // [, {
#define JP_COLN KC_QUOT // :, *
#define JP_RBRC KC_NUHS // ], }
#define JP_BSLS KC_INT1 // \, _


// Layer shorthand
enum layuer_shorthand {
  _USB = 0,
  _JPB,
  _US1,
  _JP1,
  _MUB,
  _MJB,
  _MU1,
  _MJ1,
  _MJ2,
  __CS,
  __FN,
  _FN2,
  _FN3,
  _MNT,
  _TMP,
};

// #define CK_WMH LT(__CS, KC_MHEN)
// #define CK_WHN LT(__FN, KC_HENK)
// #define CK_MMH LT(__CS, KC_LANG2)
// #define CK_MHN LT(__FN, KC_LANG1)

#define KEYS_MAX 256
#define PHYS 0
#define LOGI 1
#define TRAN 2
bool keys[KEYS_MAX][3] = {};

#define MODE_NORMAL 0
#define MODE_MAC 1
uint8_t mode = MODE_NORMAL;
uint8_t layer = _USB;
uint16_t last_press_keycode = KC_NO;

// Defines the keycodes used by our macros in process_record_user
enum custom_keycodes {
  QMKBEST = SAFE_RANGE,
  CK_WMH,
  CK_WHN,
  CK_MMH,
  CK_MHN,
};


const uint16_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {
  /* QWERTY(US)
                                                                                                                                                *
.-----------------------------------------------------------------------------------------------------------------------------------------------
```