令和６年度長野県産業教育振興会 特別生徒研究助成 報告書

研究テーマ 自作キーボードの製作

長野県上田千曲高等学校 メカニカル工学科
研究者：太田凛哉、江幡橙侍、プレブゾルボート
指導教諭：金井孝昭

１ 研究目的

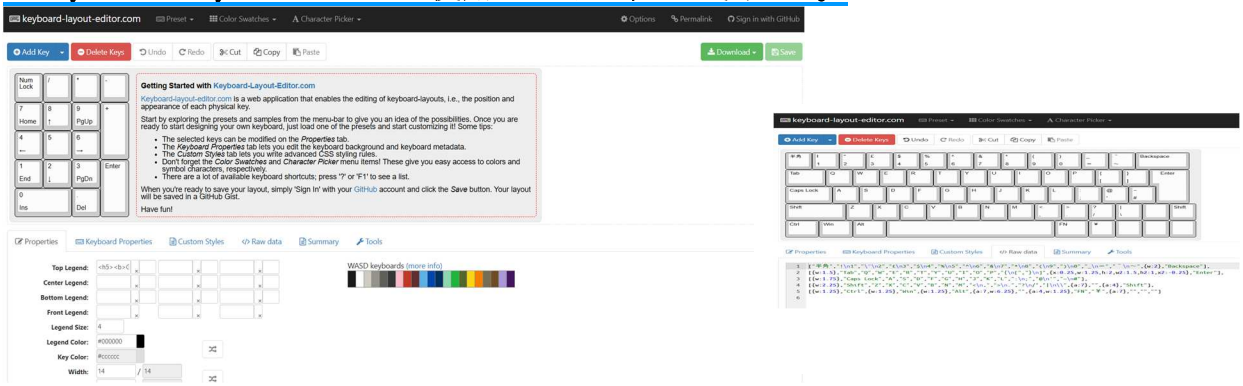
・回路図を製作し根本的な仕組みから理解する。
・PCB 基板の設計の仕方を学ぶ。
・ファームウェアを書き、実際に動作＆実用できるようにする。

２ 使用ソフト・サイト

・keyboard-layout-editor
・KiCad
・Freerouting
・JLCPCB
・QMK MSYS
・Visual Studio Code

３ 製作手順

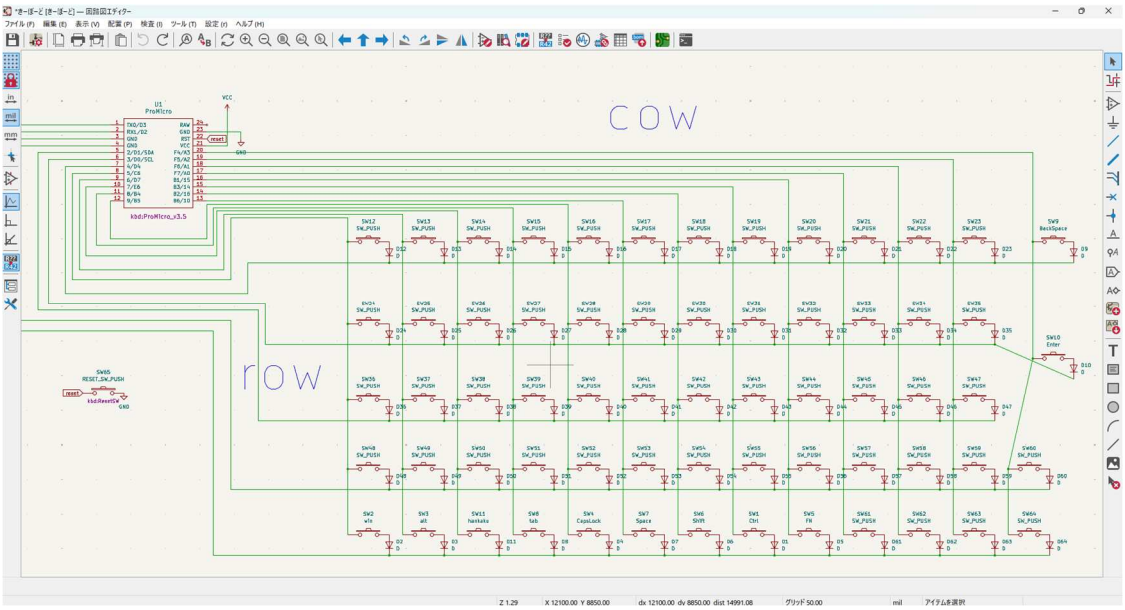
・keyboard-layout-editor を使用しレイアウトを決める。



出来上がった配列。

65%キーボードを参考に、矢印キーを内側に組み込み、半角全角キーはそのままにした。

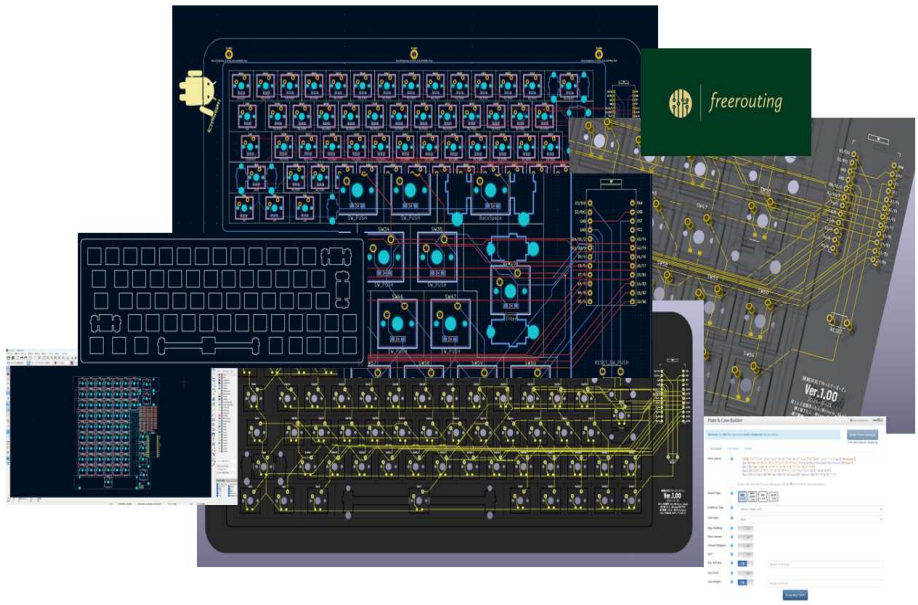
・KiCad を使い回路図を作成



部品定義を設定。
キーマトリクス方式で配線する。ダイオードも使い複数同時押しにも対応。

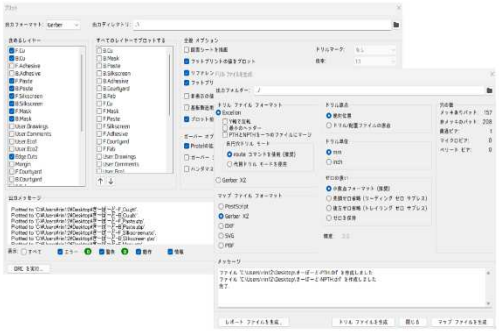
・KiCad を使い PCB 基板を設計

部品や配線の定義をしっかりと行ったあとに PlateCaseBuilder を使用しトッププレートを作成。それに合わせてフットプリントを配置。
Freerouting を使用し配線データを作成。細かい所を修正しデザイン定義を行い設計データは完成。

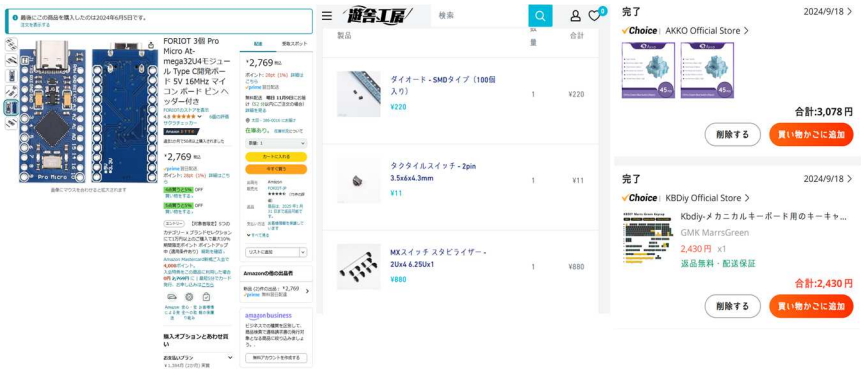


・発注&部品購入

発注を行うためにガーバーデータを生成。JLCPCB というサービスを使いメイン基板とトッププレートを発注。
基板製作代:$31.70 送料:9.58 合計:$41.28　（日本円で 6236 円）
（最低発注枚数が 5 枚なので 1 枚あたり 1250 円程）※2024/9/21

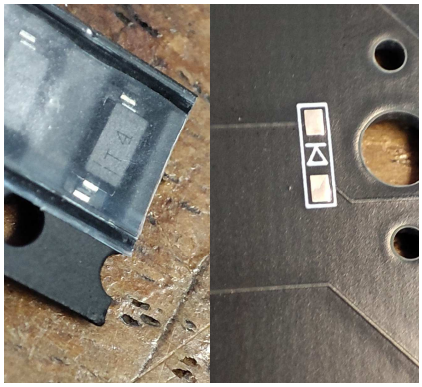


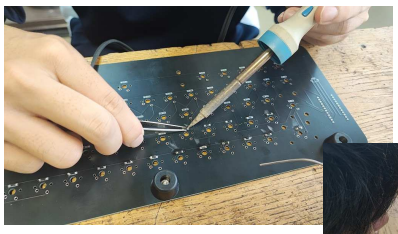
その他キーボードとして組み立てるために部品を購入。
ProMicro・ダイオード・リセットスイッチ・スタビライザー・キースイッチ・キーキャップ 約 1 万円

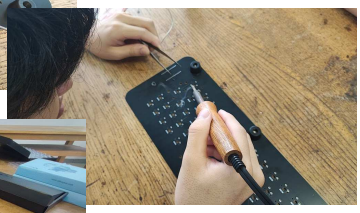


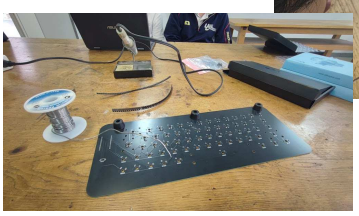
・部品実装

チップダイオードをキーの数分向きを揃えてはんだ付け。
このダイオードにより同時にキーを複数押しても正常に動作するようになる。





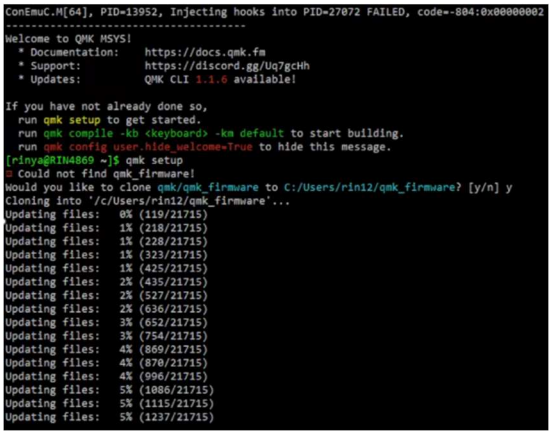


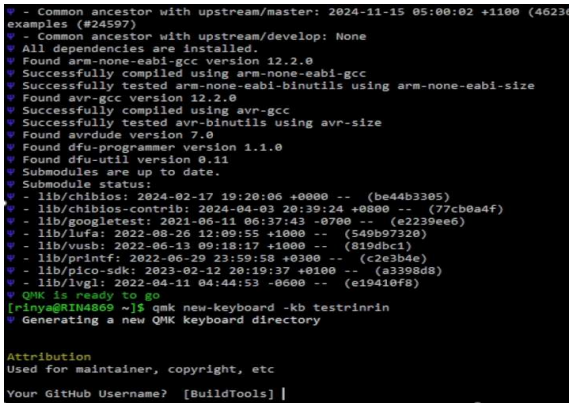


トッププレートにキースイッチをはめ込み、サイズの違うキーにはスタビライザーを取り付けたあとに、キースイッチをメイン基板にはんだ付けする。
キーキャップや部品を取り付けたらハードウェア部分は完成。

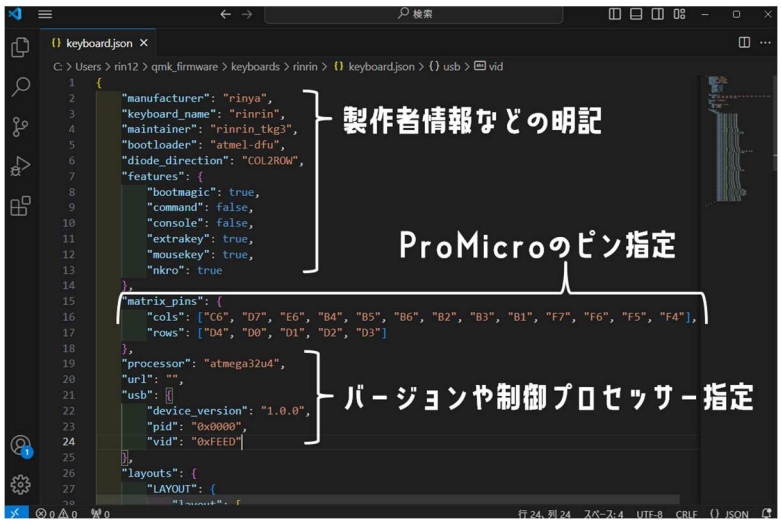
・ファームウェア実装

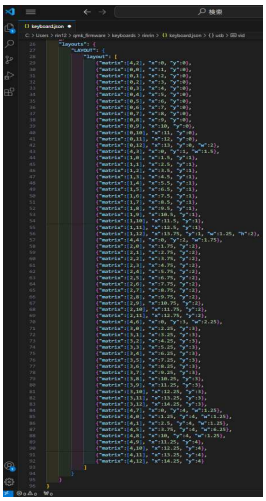
使用ファームウェアは QMK Firmware まずは QMK MSYS をダウンロードし、セットアップコマンドを走らせてその後に新しいキーマップを作成するコマンドを実行する。



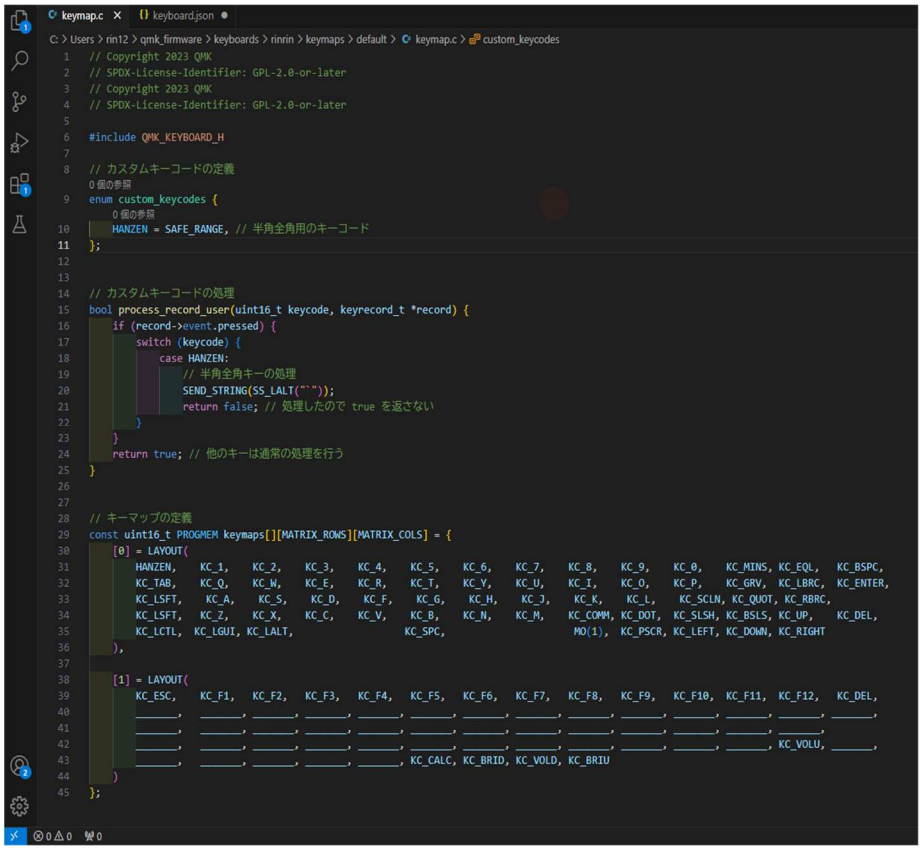


設定用.json ファイルを編集し、ProMicro のピン指定やキーの座標データ、他にも様々な情報を入力する。





次にキーマップの定義を行う。
これはこのキーをなんの入力キーにするかの設定やレイヤー・マクロの定義などを行うかを定義する C#のコードになる。



```c
// Copyright 2023 QMK
// SPDX-License-Identifier: GPL-2.0-or-later
// Copyright 2023 QMK
// SPDX-License-Identifier: GPL-2.0-or-later

#include QMK_KEYBOARD_H

// カスタムキーコードの定義
0 個の参照
enum custom_keycodes {
    0 個の参照
    HANZEN = SAFE_RANGE, // 半角全角用のキーコード
};


// カスタムキーコードの処理
bool process_record_user(uint16_t keycode, keyrecord_t *record) {
    if (record->event.pressed) {
        switch (keycode) {
            case HANZEN:
                // 半角全角キーの処理
                SEND_STRING(SS_LALT("`"));
                return false; // 処理したので true を返さない
        }
    }
    return true; // 他のキーは通常の処理を行う
}


// キーマップの定義
const uint16_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {
    [0] = LAYOUT(
        HANZEN,   KC_1,   KC_2,   KC_3,   KC_4,   KC_5,   KC_6,   KC_7,   KC_8,   KC_9,   KC_0,   KC_MINS, KC_EQL,  KC_BSPC,
        KC_TAB,   KC_Q,   KC_W,   KC_E,   KC_R,   KC_T,   KC_Y,   KC_U,   KC_I,   KC_O,   KC_P,   KC_GRV,  KC_LBRC, KC_ENTER,
        KC_LSFT,   KC_A,   KC_S,   KC_D,   KC_F,   KC_G,   KC_H,   KC_J,   KC_K,   KC_L,   KC_SCLN, KC_QUOT, KC_RBRC,
        KC_LSFT,  KC_Z,   KC_X,   KC_C,   KC_V,   KC_B,   KC_N,   KC_M,   KC_COMM, KC_DOT,  KC_SLSH, KC_BSLS, KC_UP,   KC_DEL,
        KC_LCTL,  KC_LGUI, KC_LALT,                KC_SPC,                 MO(1),   KC_PSCR, KC_LEFT, KC_DOWN, KC_RIGHT
    ),

    [1] = LAYOUT(
        KC_ESC,   KC_F1,   KC_F2,   KC_F3,   KC_F4,   KC_F5,   KC_F6,   KC_F7,   KC_F8,   KC_F9,   KC_F10,  KC_F11,  KC_F12,  KC_DEL,
        _______,  _______, _______, _______, _______, _______, _______, _______, _______, _______, _______, _______, _______, _______,
        _______,  _______, _______, _______, _______, _______, _______, _______, _______, _______, _______, _______, _______,
        _______,  _______, _______, _______, _______, _______, _______, _______, _______, _______, _______, KC_VOLU, _______,
        _______,  _______, _______, _______, _______, KC_CALC, KC_BRID, KC_VOLD, KC_BRIU
    )
};
```

できた 2 つのファイルをコンパイルして.hex ファイルが生成されたらファームウェア自体は完成となる。

出来上がったファームウェアを ProMicroWebUpdater を使い、書き込む。

・完成

動作確認を行い無事動いたのでこれで完成となる。

4　考察・反省感想

・基板設計や実装についてしっかりと理解する事ができた。

・時間に余裕を持って作業することができたので、特にミスもなく無事完成させることができた。

・ファームウェアなどはまだ十分改善できるので、今後もアップデートを行っていきたい。

・１年を通して１つの周辺機器としてしっかりと実用的に動作するものを作れてとても良かったと感じた。