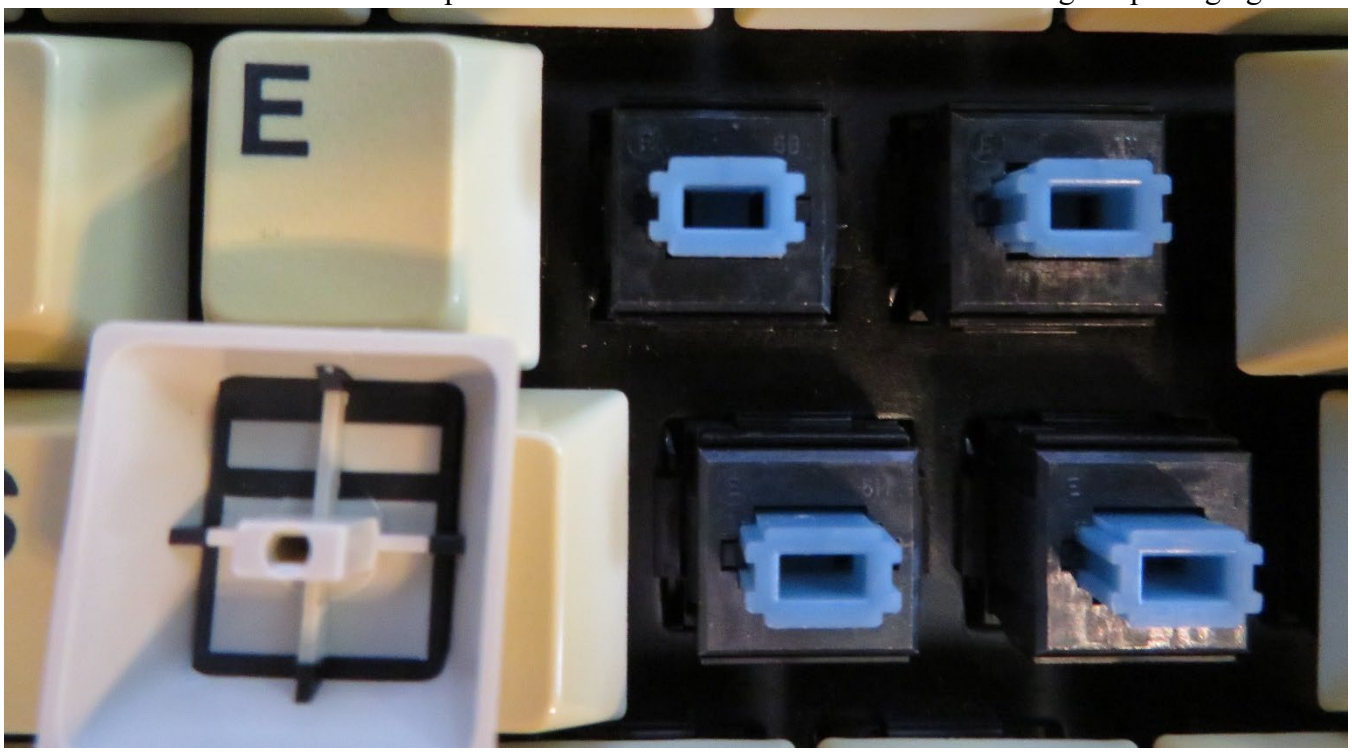
Copam K-430 Alps Electric USB Hotswap Retrofit PCB Project

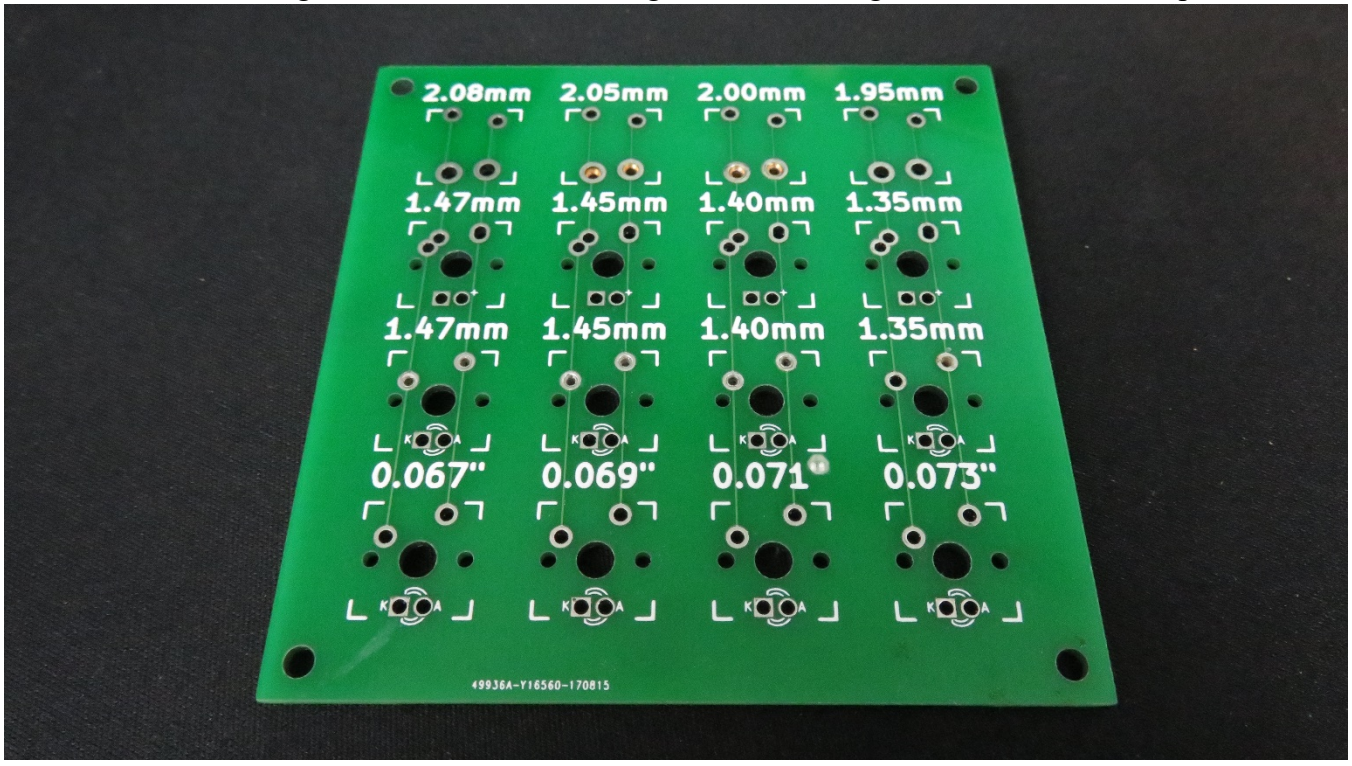
Andrew B. Davis

After some time searching for Blue Alps, a highly desirable switch manufactured in the late 1980s by Alps Electric, I finally sourced a keyboard that I suspected would contain them. The only record of this mysterious ‘Copam’ model on the internet was from a user on a computer forum who said he sold it since it used a non-standard protocol. I took a chance since it was still in original packaging.

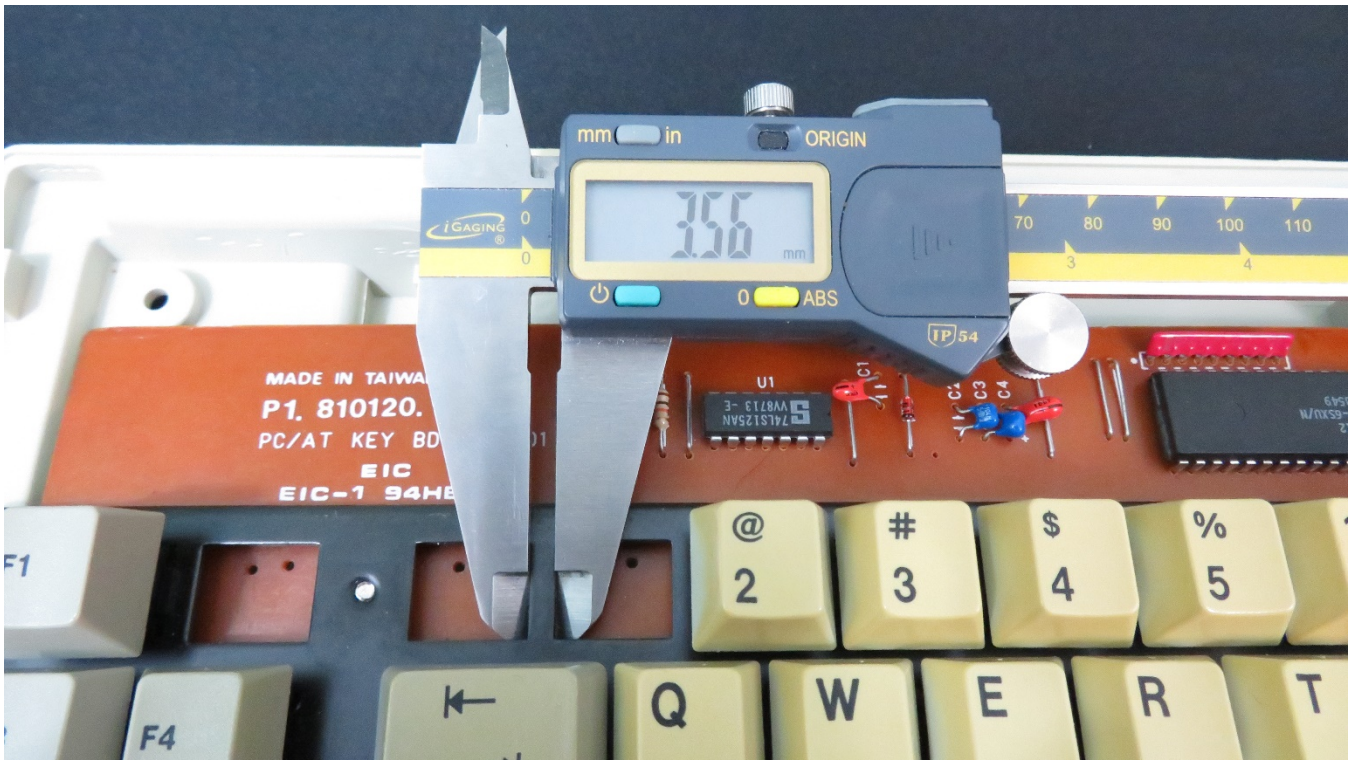


Success, it indeed contain Blue Alps! The keyboard felt great to type on but using my DIN-5 AT/XT converter yielded nothing but error codes.

After sitting in my closet for almost a year, I decided that I liked this yellowed dinosaur enough to teach myself KiCad and design a PCB (printed circuit board) that uses press-fit TE Connectivity Holtite sockets. The benefits include full reprogrammability, native USB compatibility and less heat stress on the delicate switch legs that comes with soldering and de-soldering should I decide to swap them out.

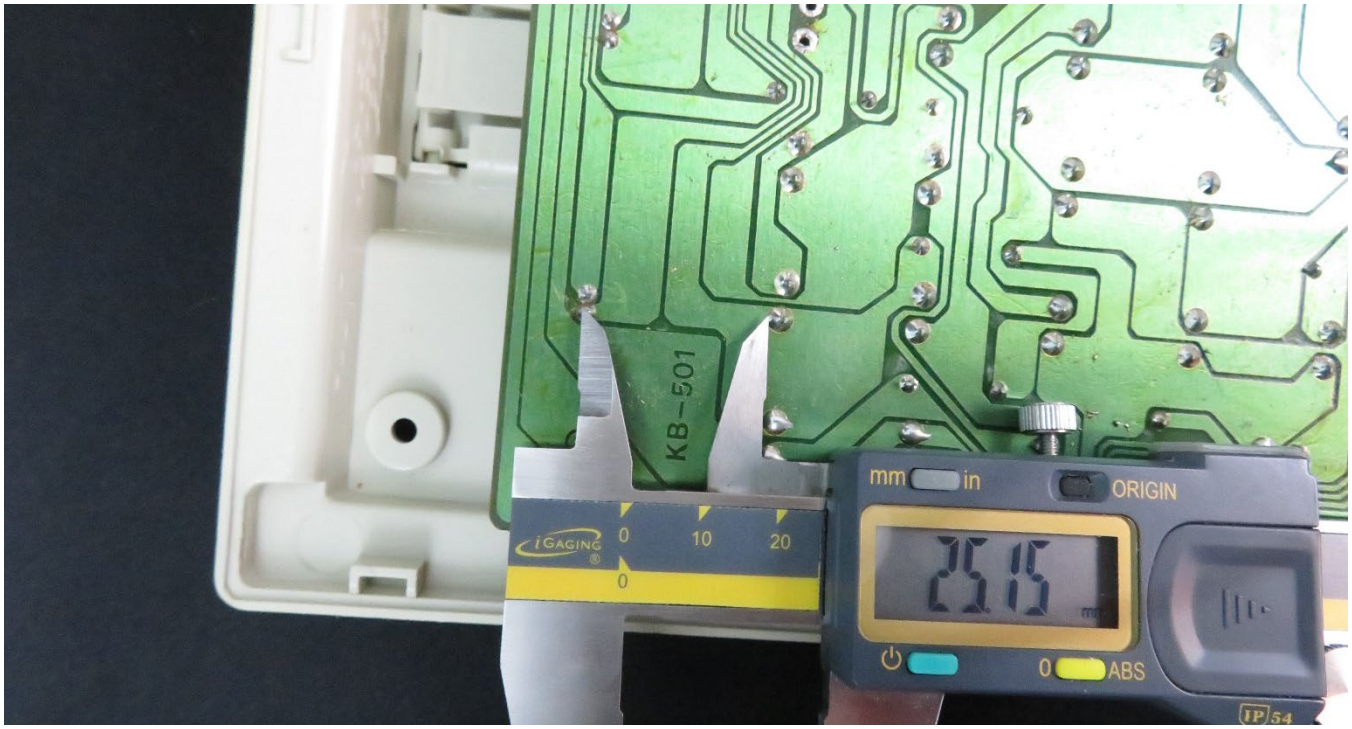


The Holtite connector spec-sheet recommended a mounting hole diameter of 2.08mm ±0.05mm so I had a friend of mine order me a prototype PCB to test a few different drill sizes. I settled on 2.00mm.

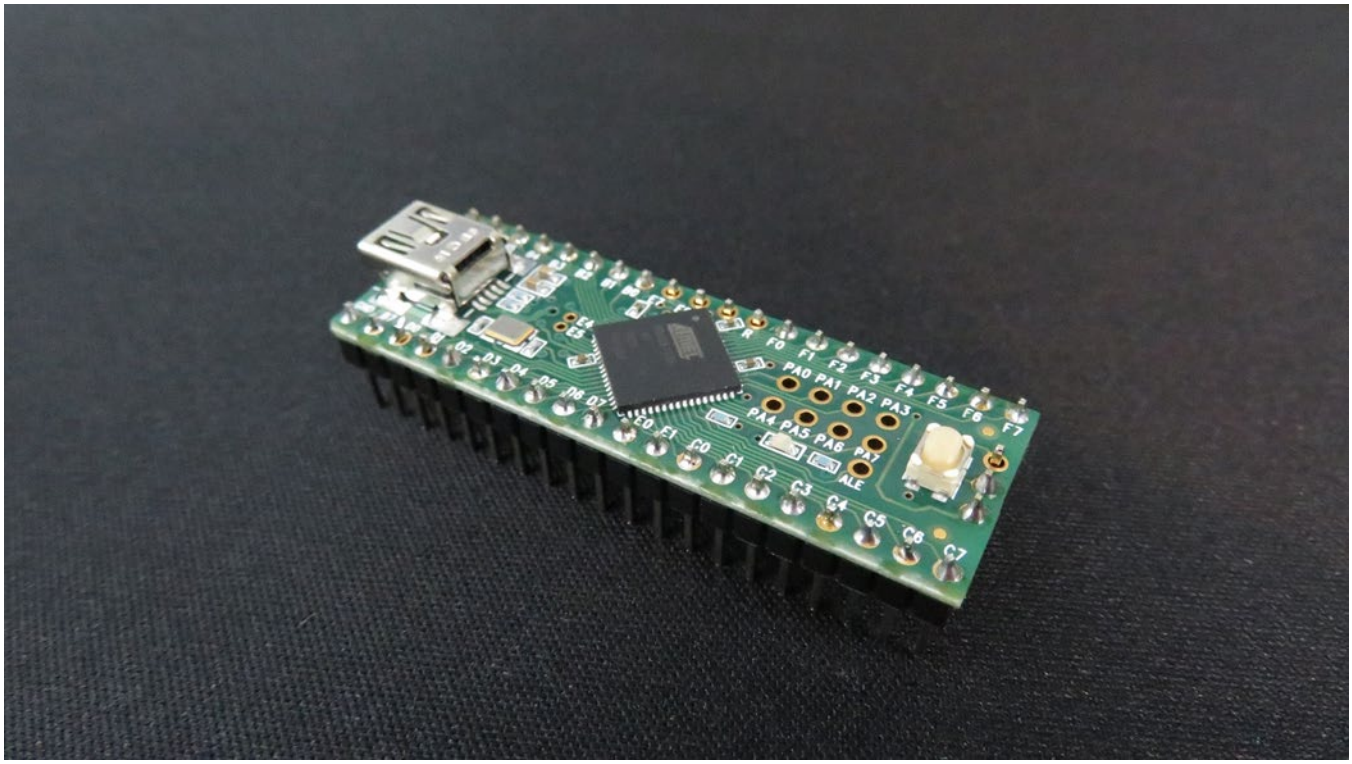


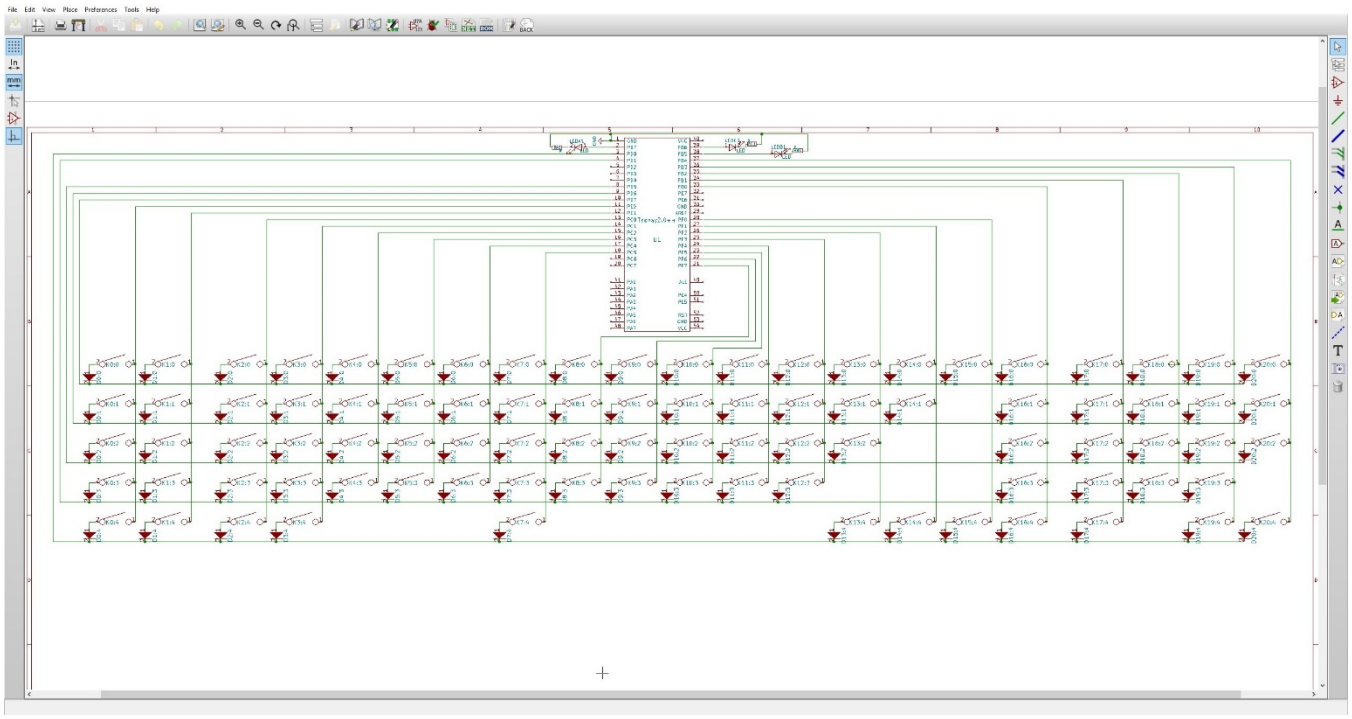
Even today's modern keyboards follow the widely adopted IBM standard of 0.75" or 19.05mm from center-to-center of each (1 unit) key. The difficult part was determining the spacing from the function keys and numpad area to the middle portion. Being off by 0.5mm could render the PCB unusable.

After bit of measuring, I was able to determine the additional width of ~9.5mm amounted to a standard separation of 1.5u or 28.575mm.

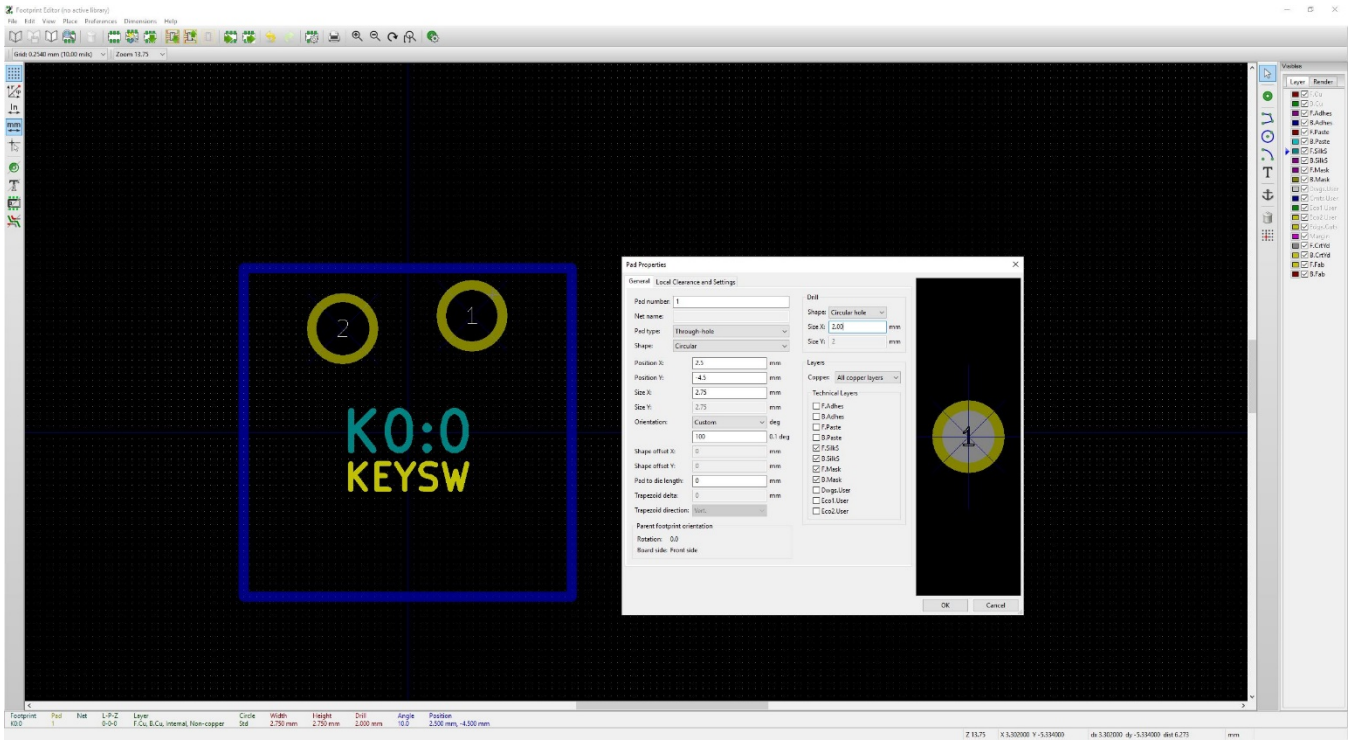


Measuring where the new lock LEDs should be located to line up with the current case.
(Picture for illustrative purposes)

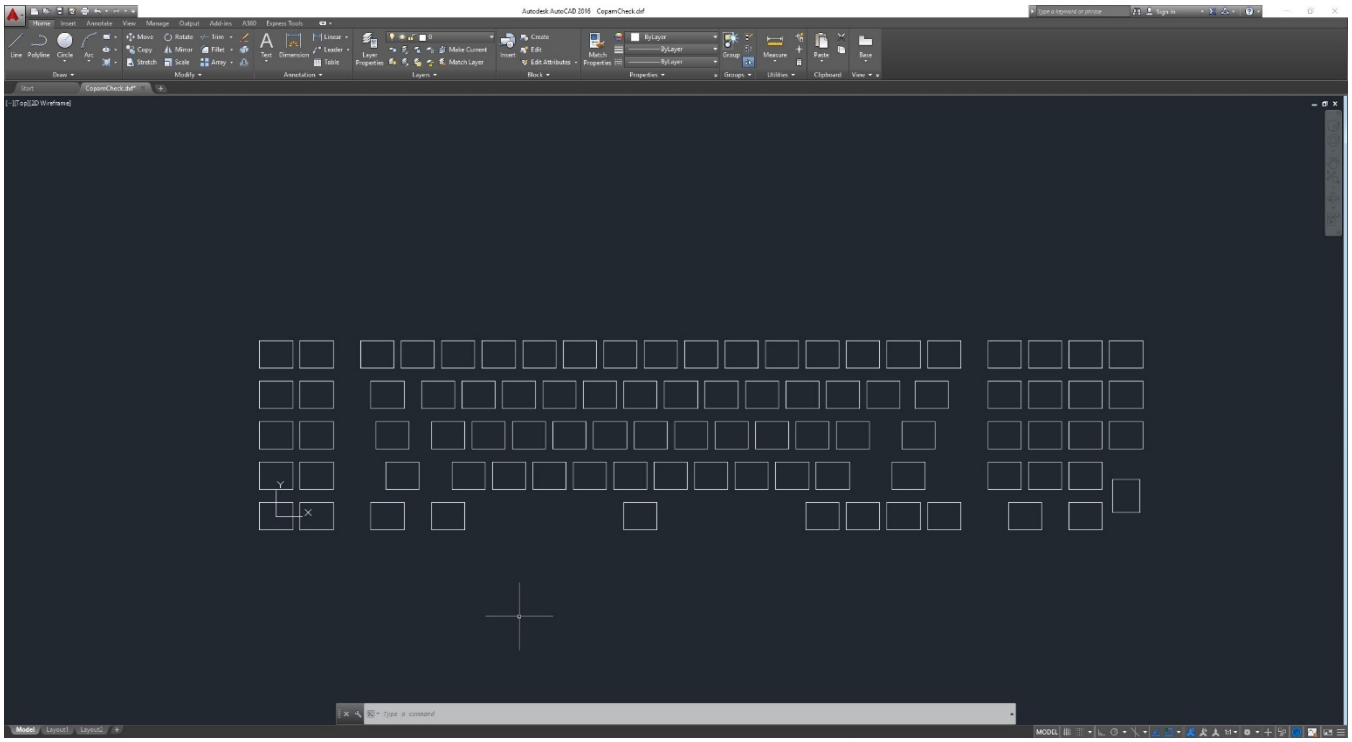

For the brain of this project, I selected the PJRC Teensy 2++ as it utilizes the AT90USB1286 chip and has more I/O and EEPROM than the more commonly used ATMEGA32U4 powered Teensy 2.0.



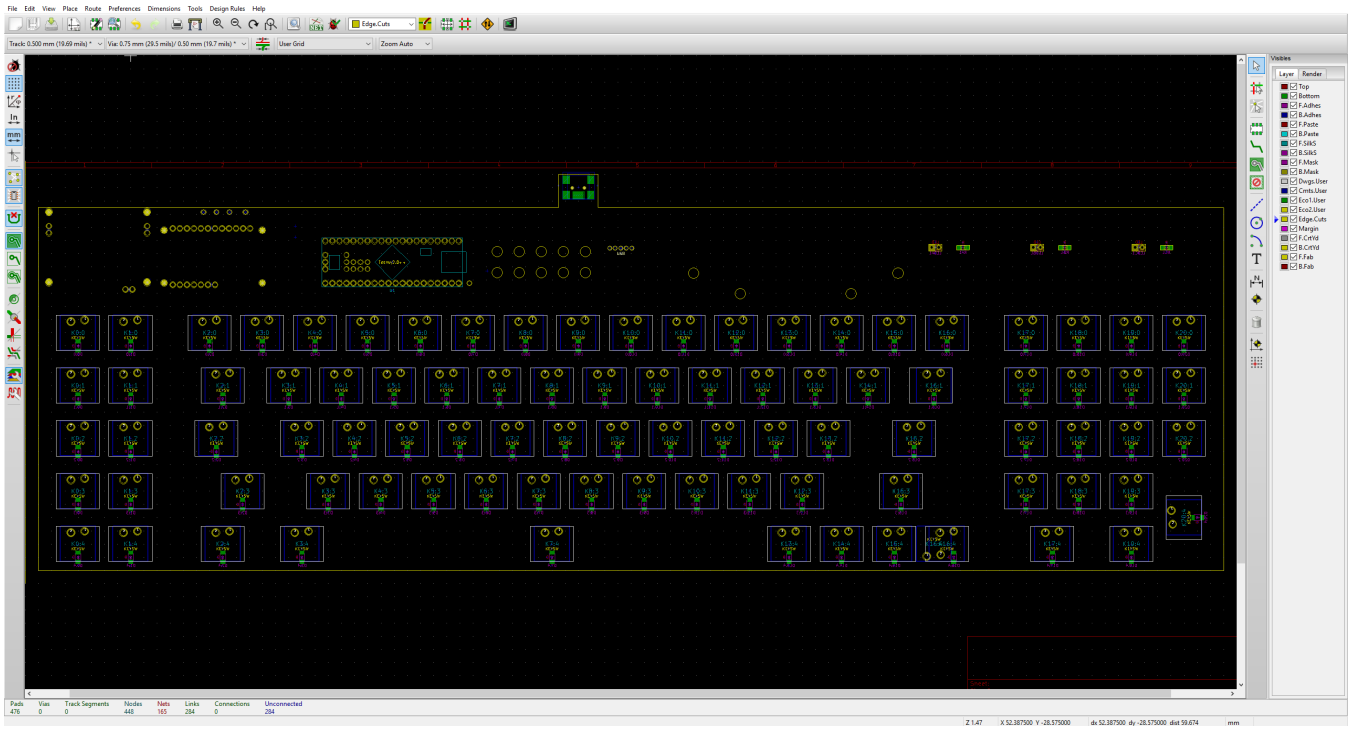
The first step in KiCad was laying out the component schematic in Eeschema and associating each drawing with their corresponding footprints. The resulting .net file will be imported into Pcbnew.



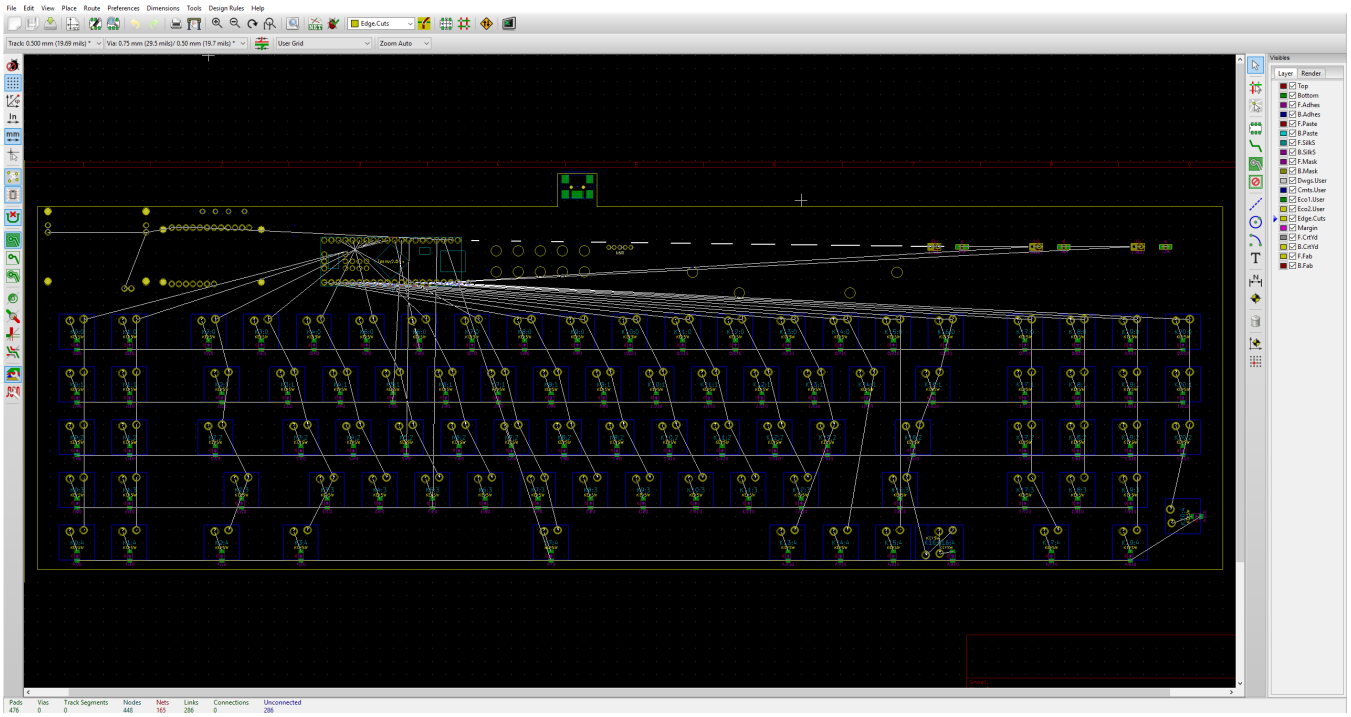
Using Footprint Editor, created a footprint for each switch using retooled Alps specification sheets from Matias and widening the drill size to 2.00mm. The footprints for diodes, resistors and LEDs were taken from the default KiCad library.



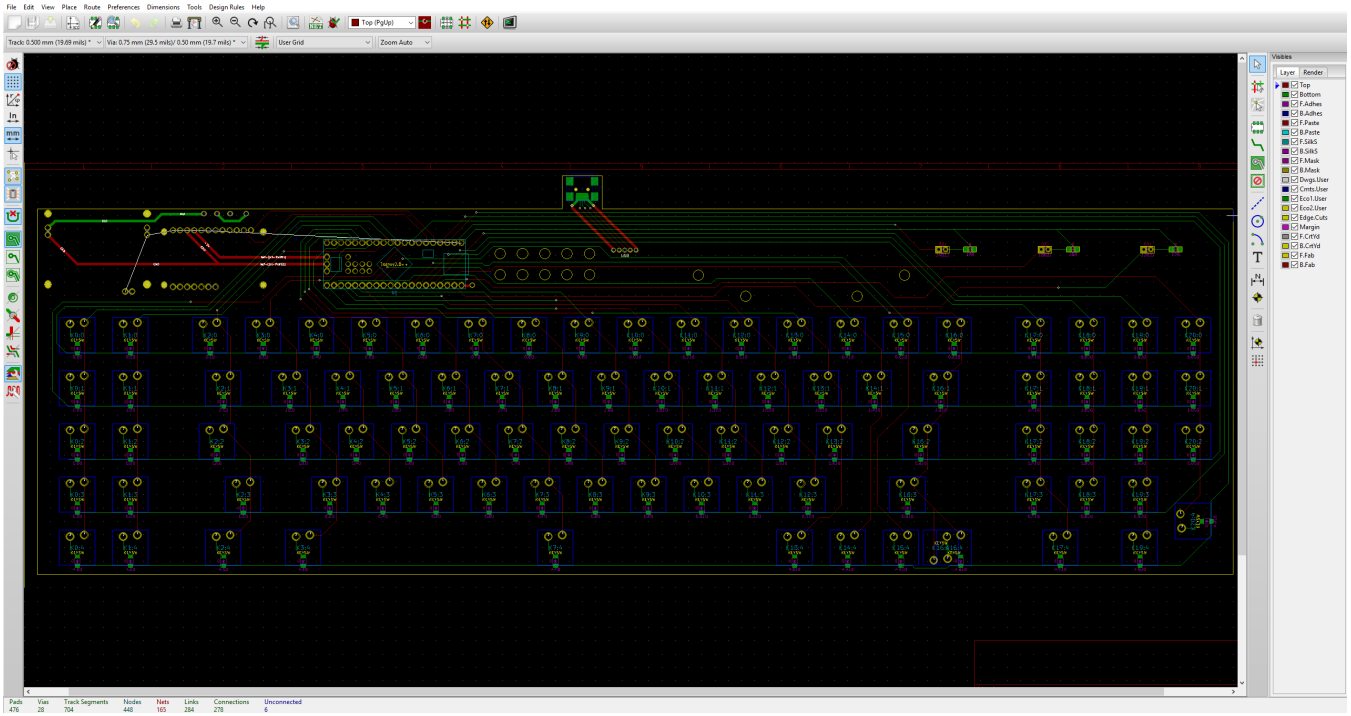
To double check my switch spacing, a basic 2D .dxf file was drawn up in AutoCAD then imported as an overlay into KiCad.



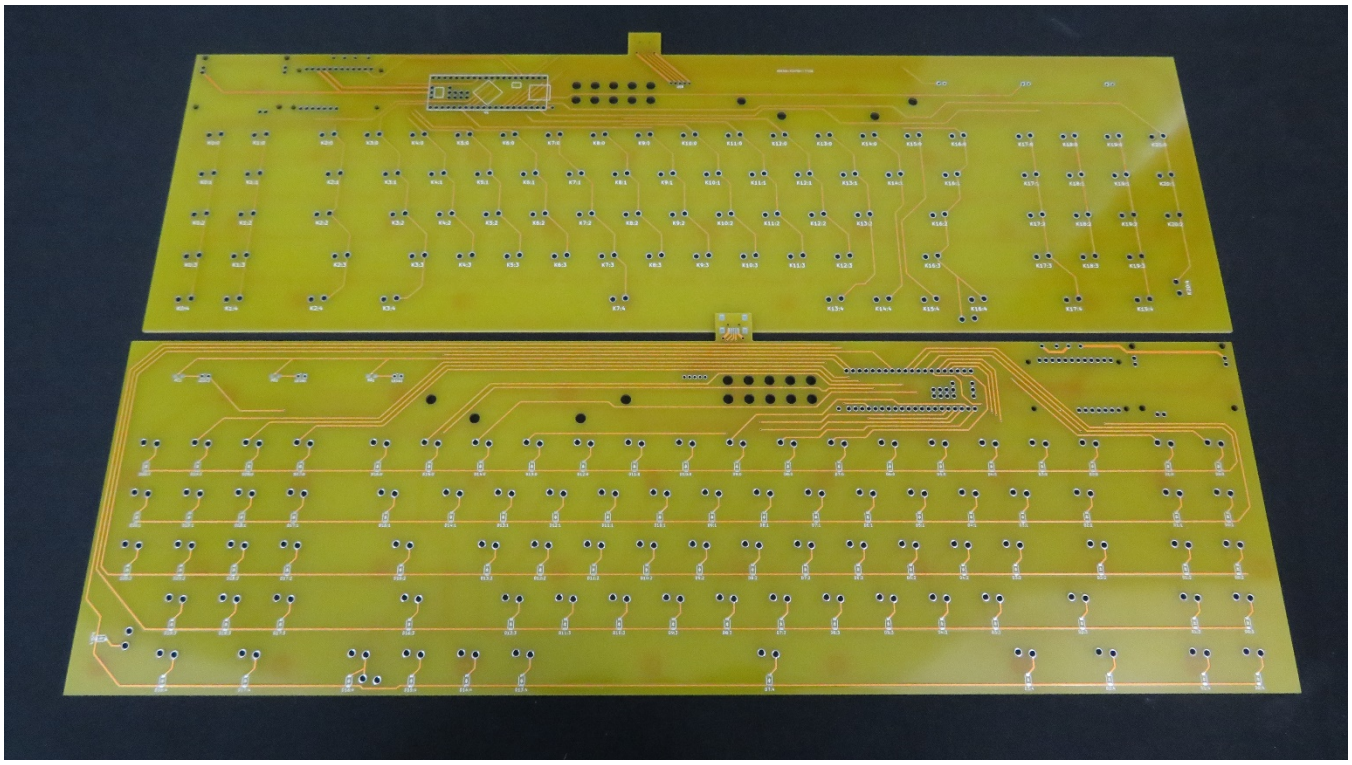
I knew the Teensy would be removed and reinstalled periodically, so its footprint was modified to use smaller Holtite sockets. Provisions were also made so the USB cable would be secured, and Bluetooth could be added in the future. Later, I learned of similar keyboards were being found using a different case design, presumably manufactured by the same OEM, so additional support was added to accommodate their modified layouts.



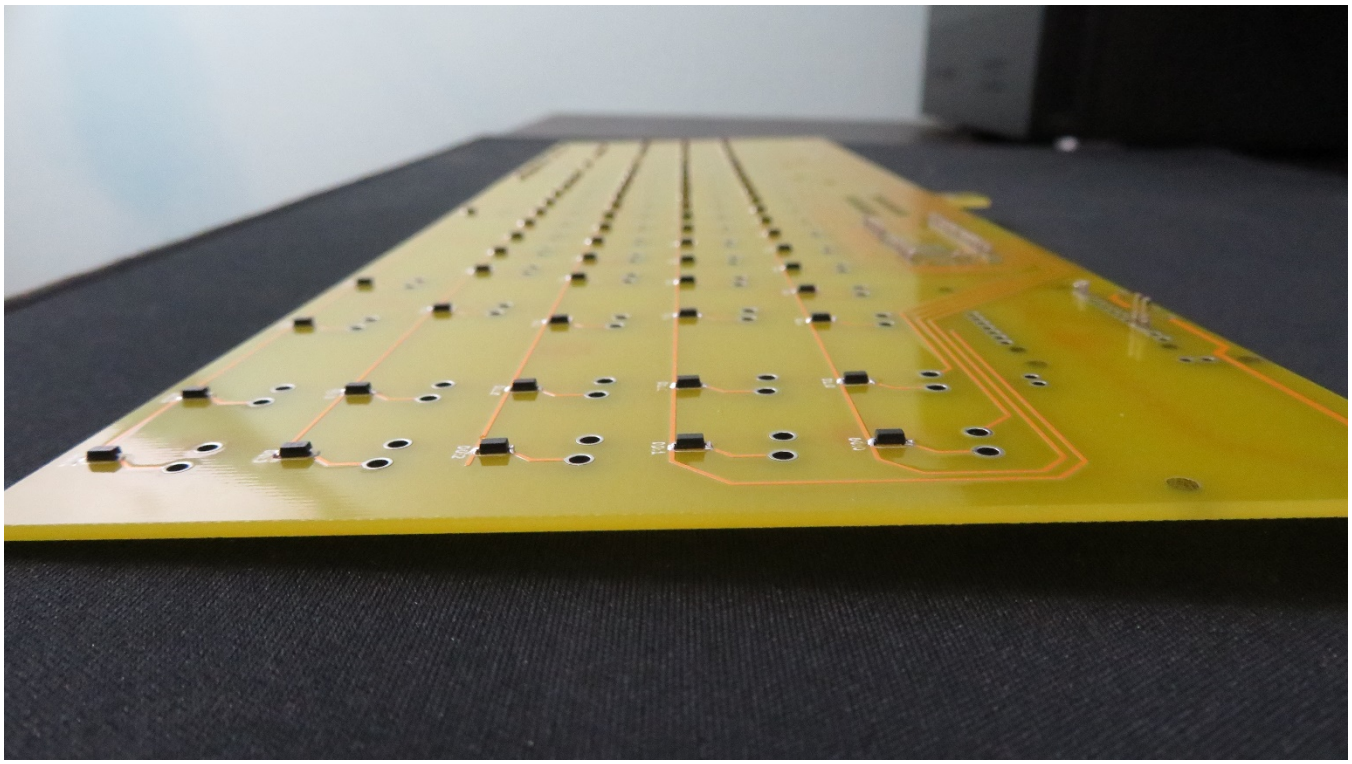
Importing the .net file from Eeschema will create lines that indicate the needed path of each trace.



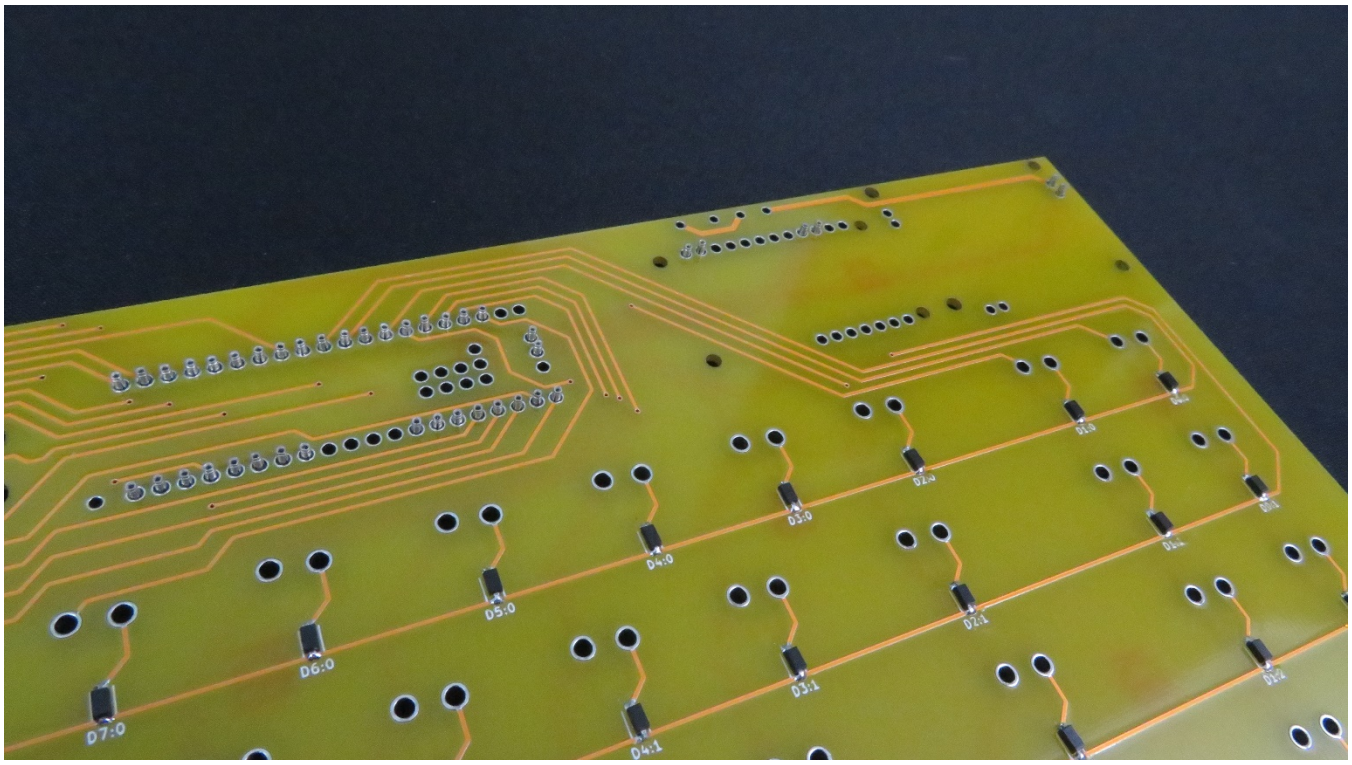
Very happy with how my manually routed traces came out as the automatic tool was a tad messy. The red traces correspond with the top layer and the green traces correspond with the back layer.



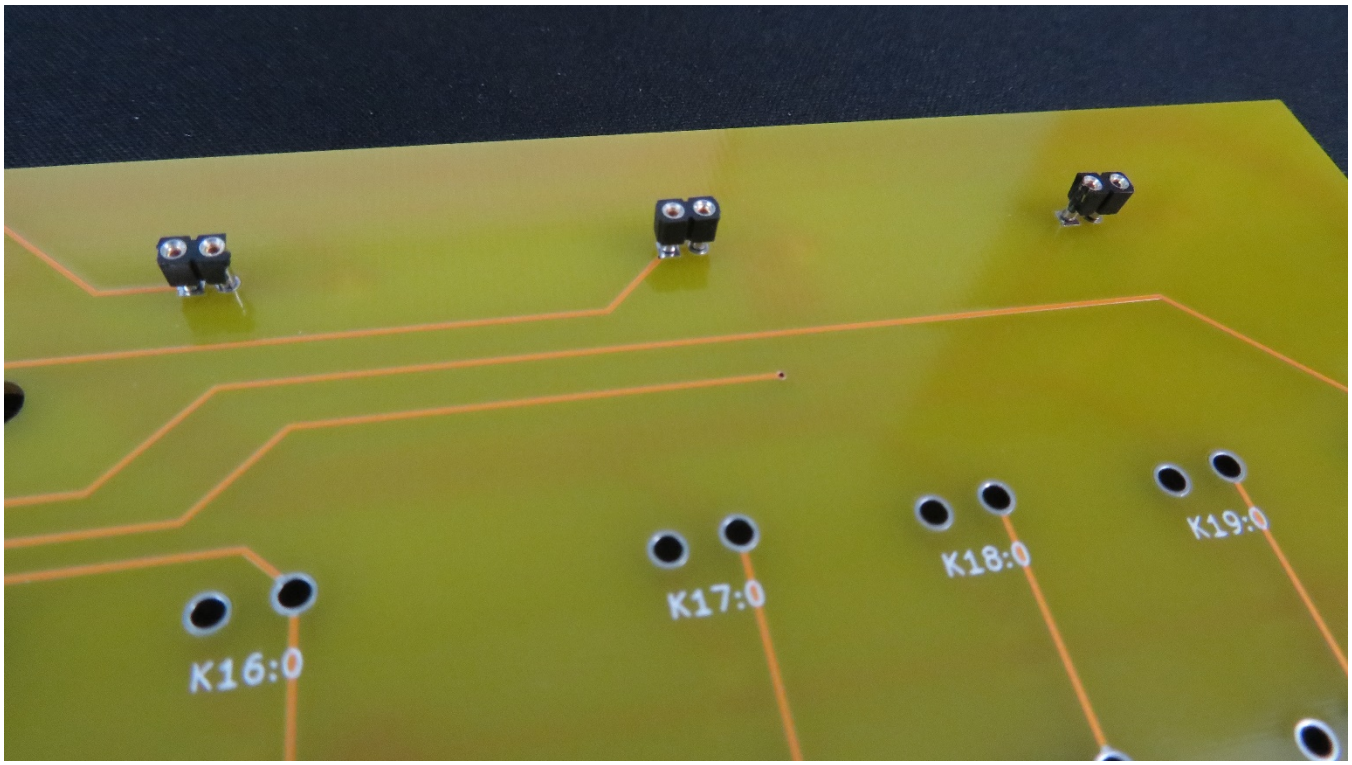
PCB have arrived, they yellow solder mask was chosen since it seemed more fitting for this retro-fit project than something like blue or red.



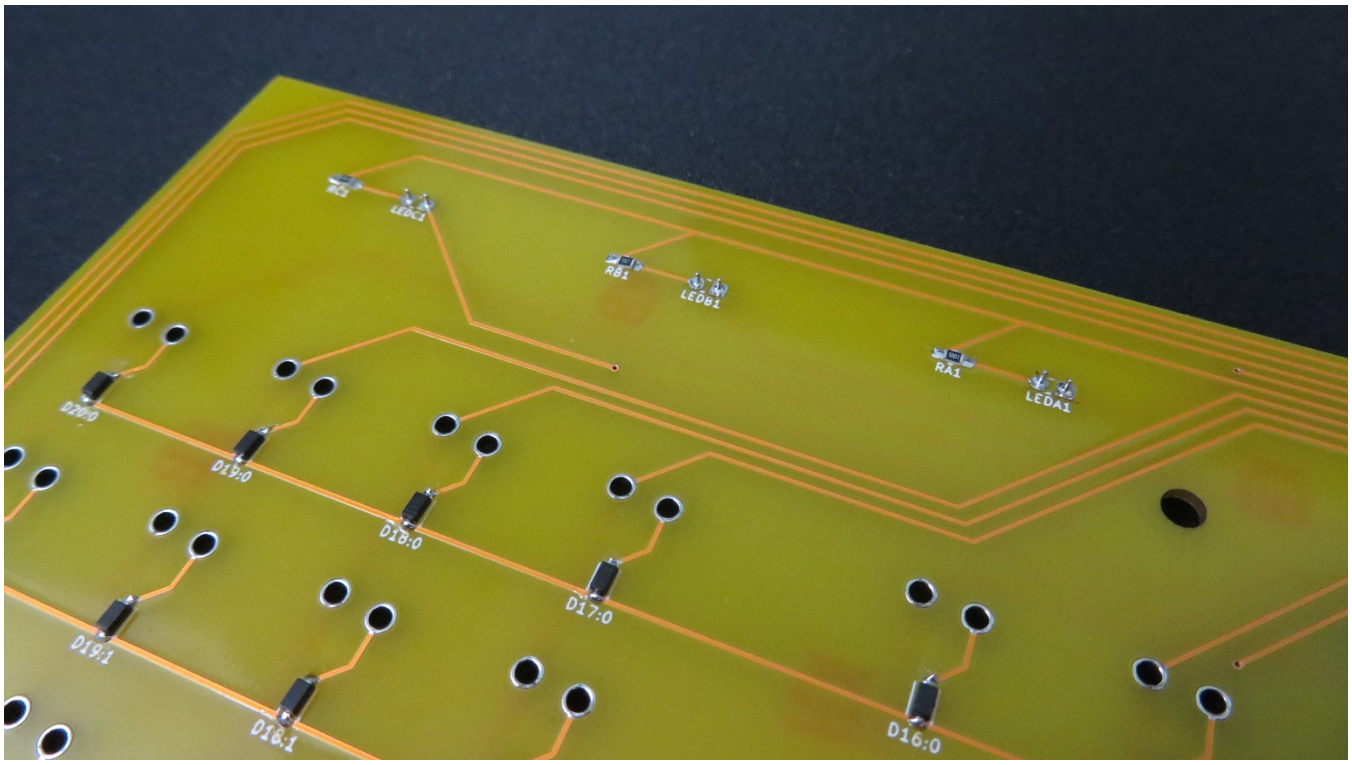
All 89 SMD 1n4148 diodes soldered in and tested with a multimeter to ensure they are all orientated correctly since unlike working with resistors, diode polarity is essential.



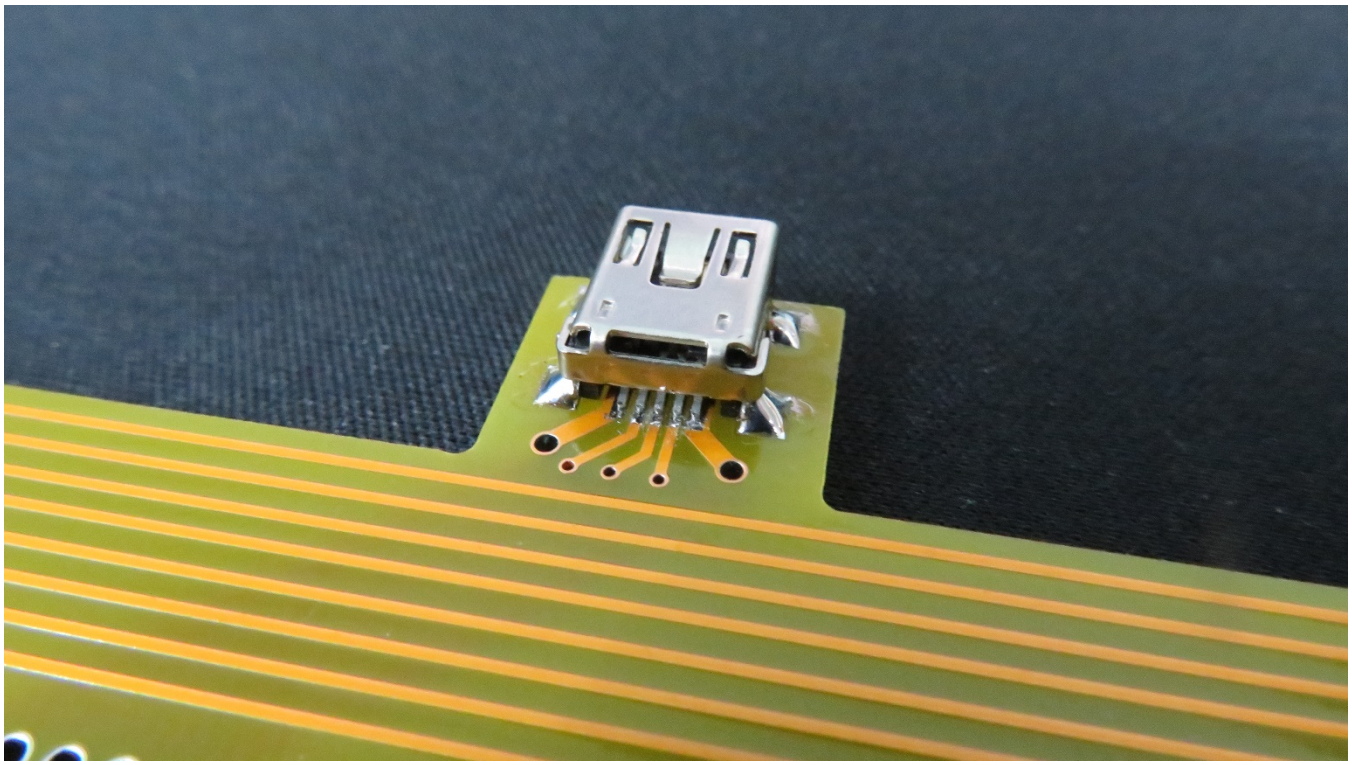
Installation of TE Holtite sockets to make the Teensy 2++ hot-swappable



LED SIP sockets soldered in for the lock lights

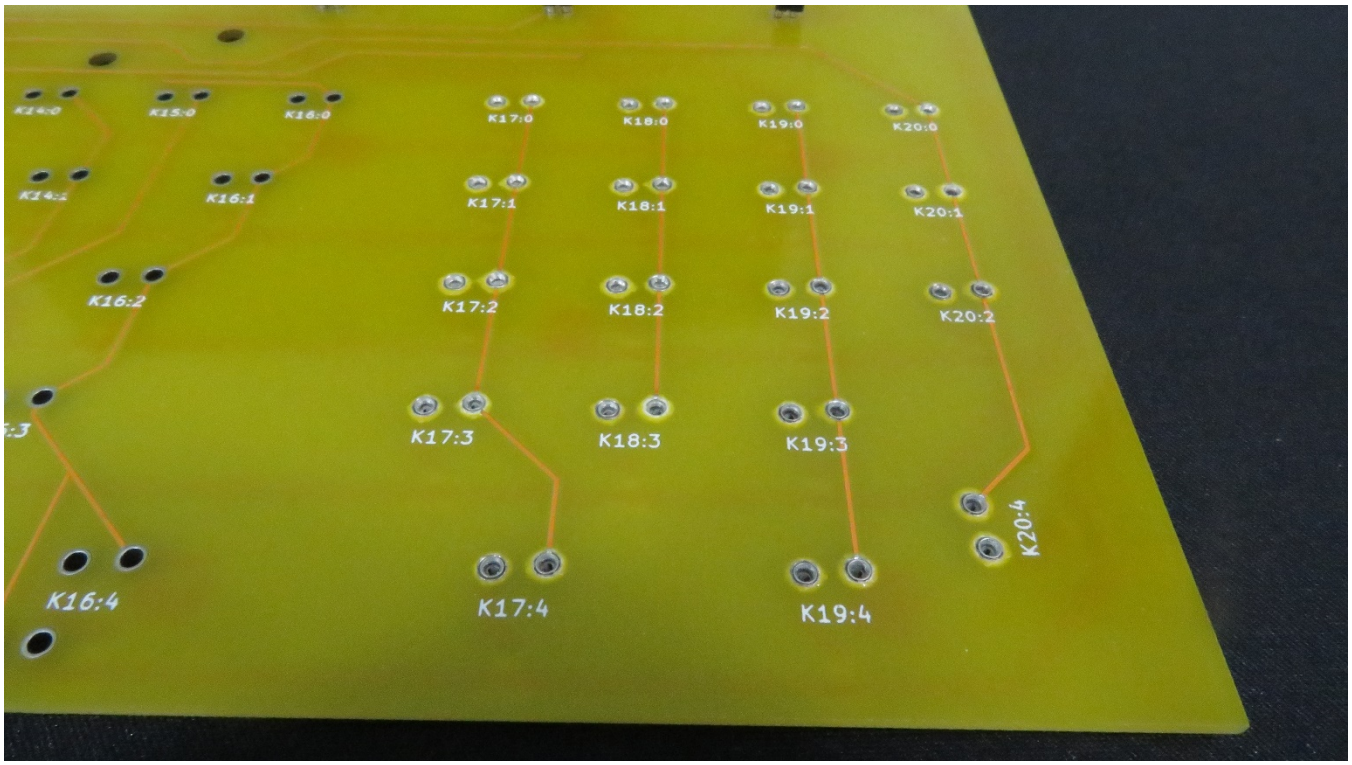


Thankfully there were only three 805 SMD resistors to solder in, I didn't realize they were so small.



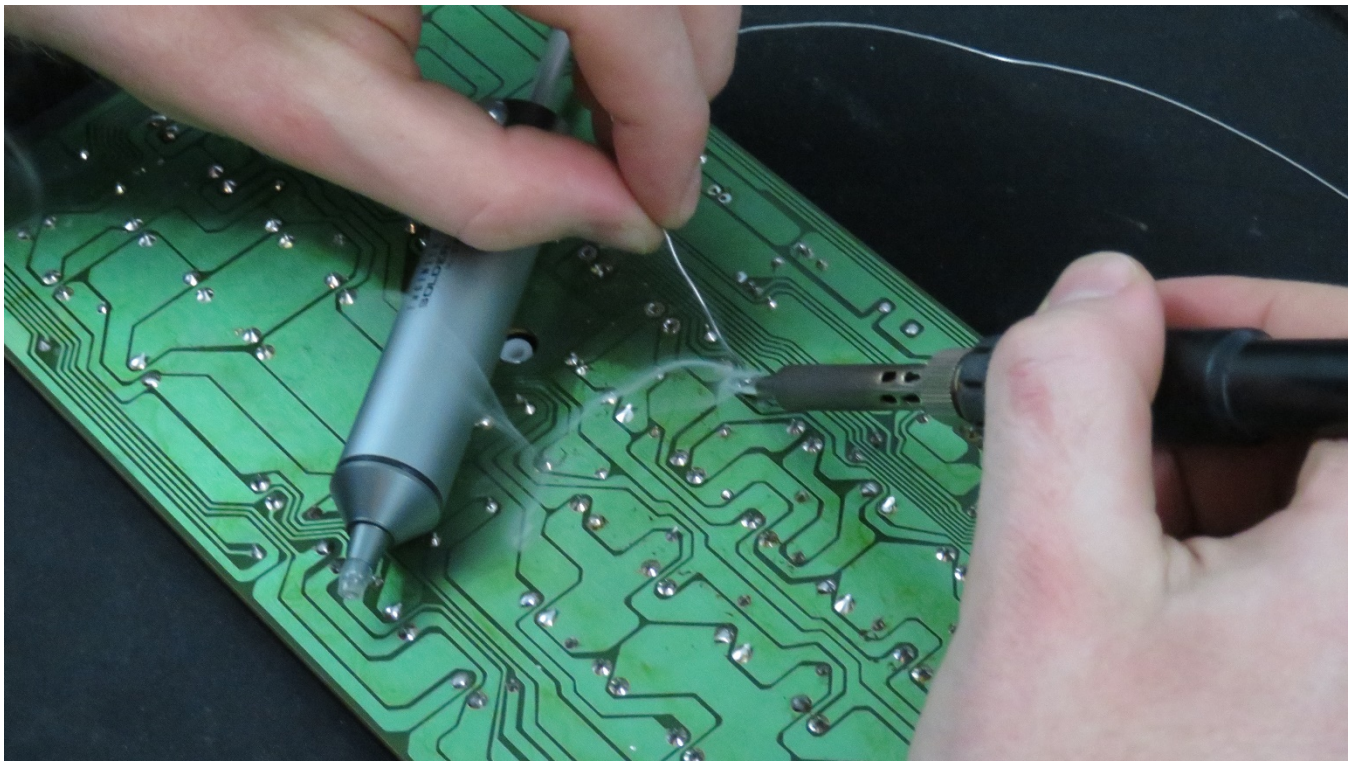
Even though I intended on using a non-removable cable, this was the point where the USB mini connector would be most accessible for soldering.

Starting the switch socket installation with the number pad area.

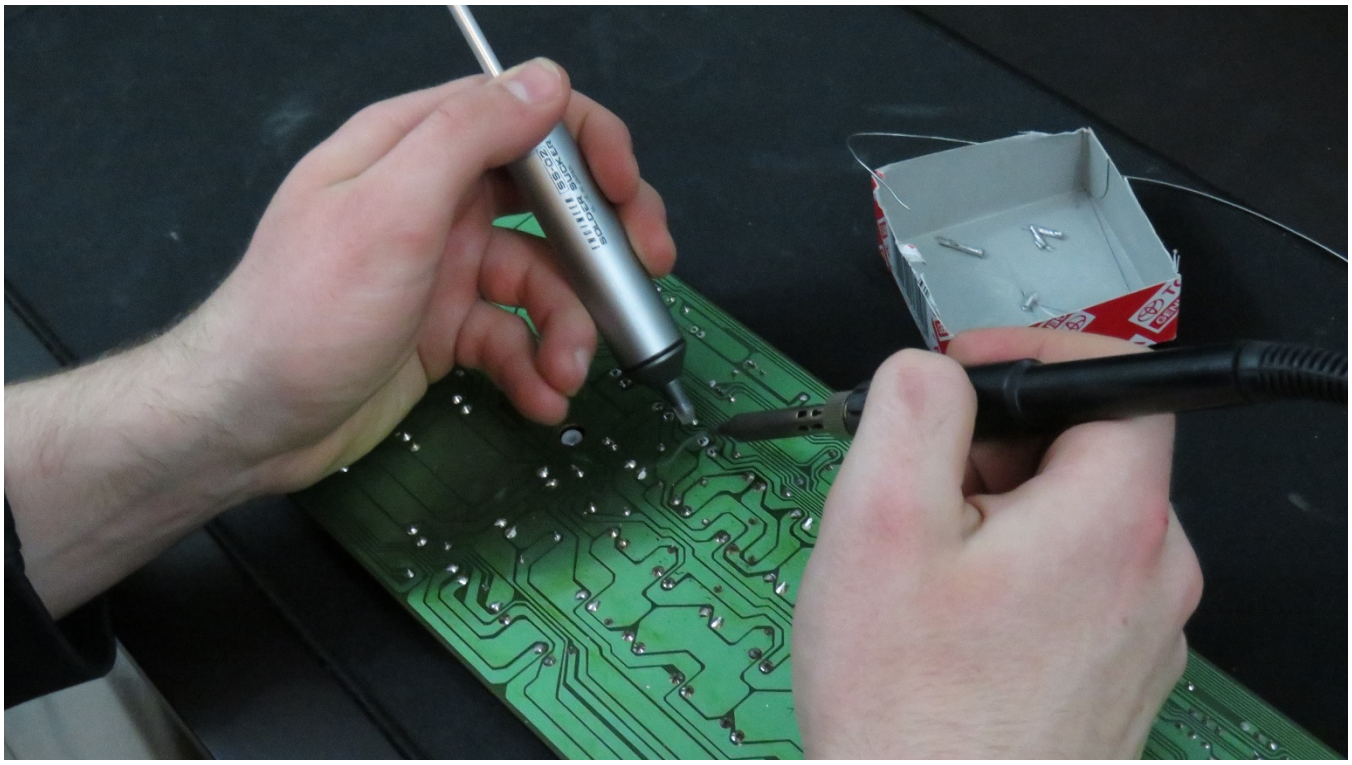


The heat applied from the soldering iron has discolored the area around the solder pads. They fit very snug, if I were to re-order from this manufacturing house the 2.05mm drill size would be a better choice.

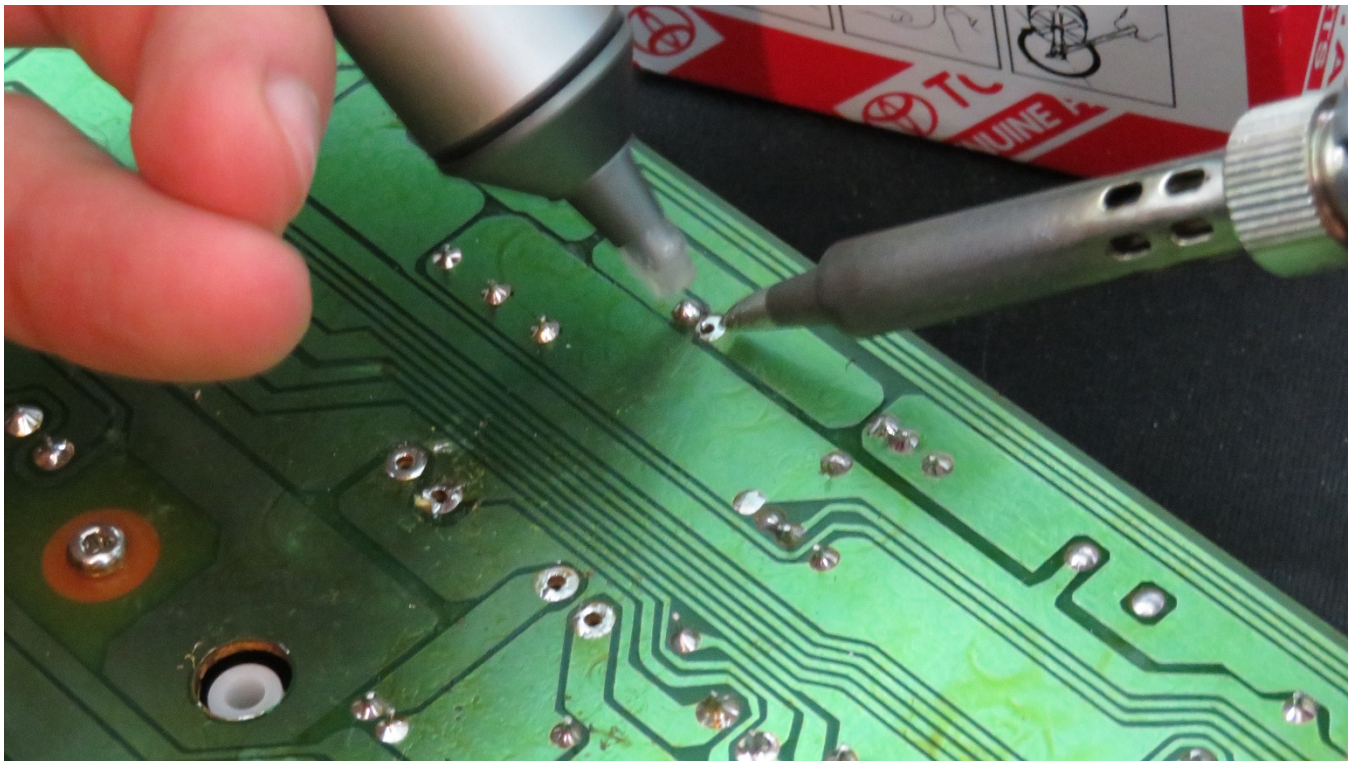
In preparation to desolder the switches, all keycaps were removed.

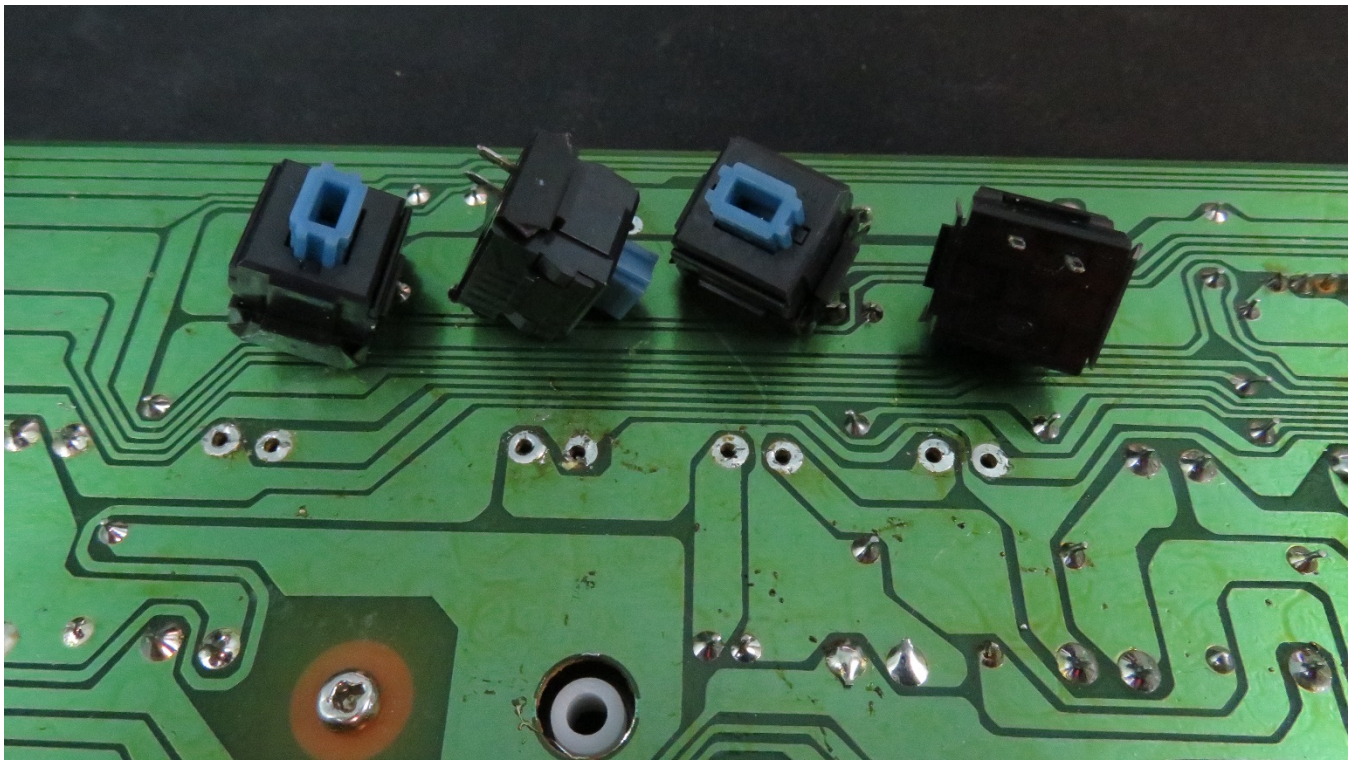


Adding fresh 63-37 rosin core solder to loosen up what has been sitting for nearly 30 years.

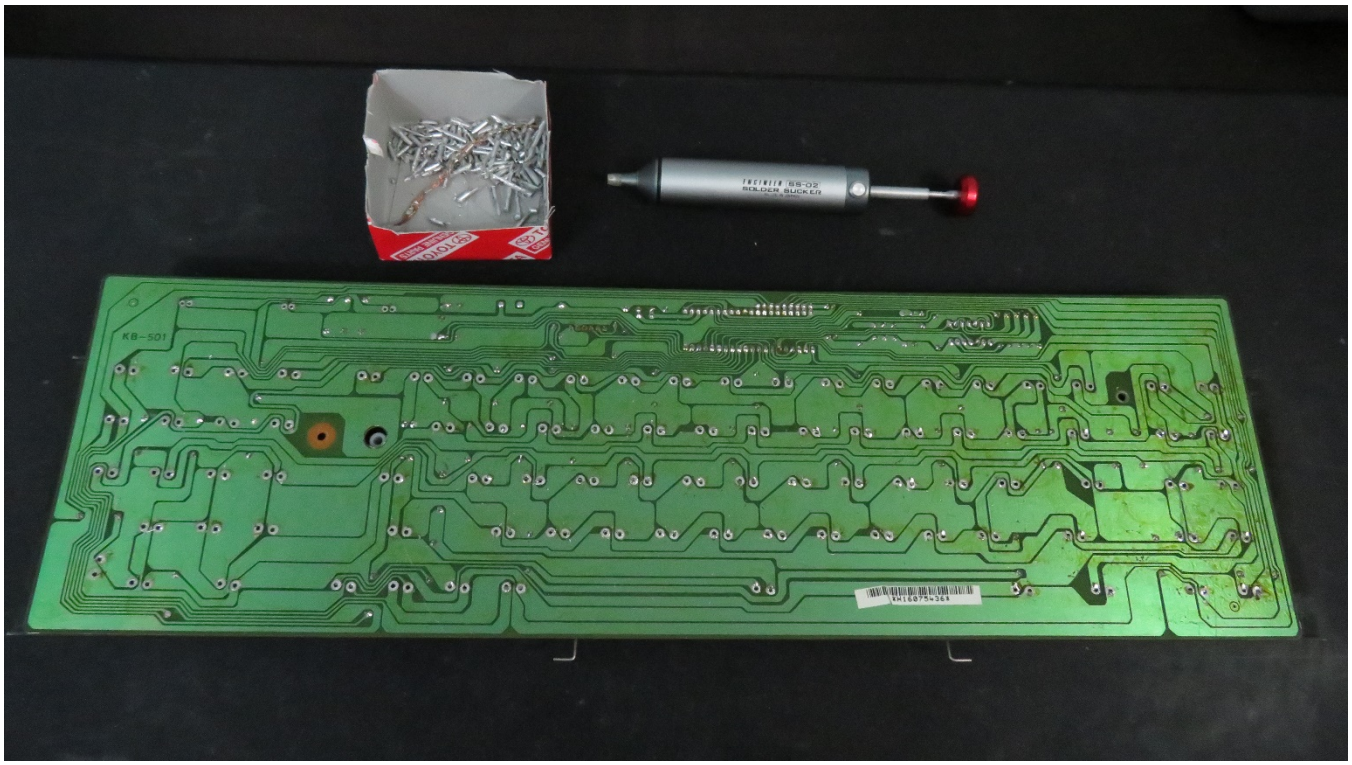


The metal sucker is spring loaded and creates a suction force to remove the solder.



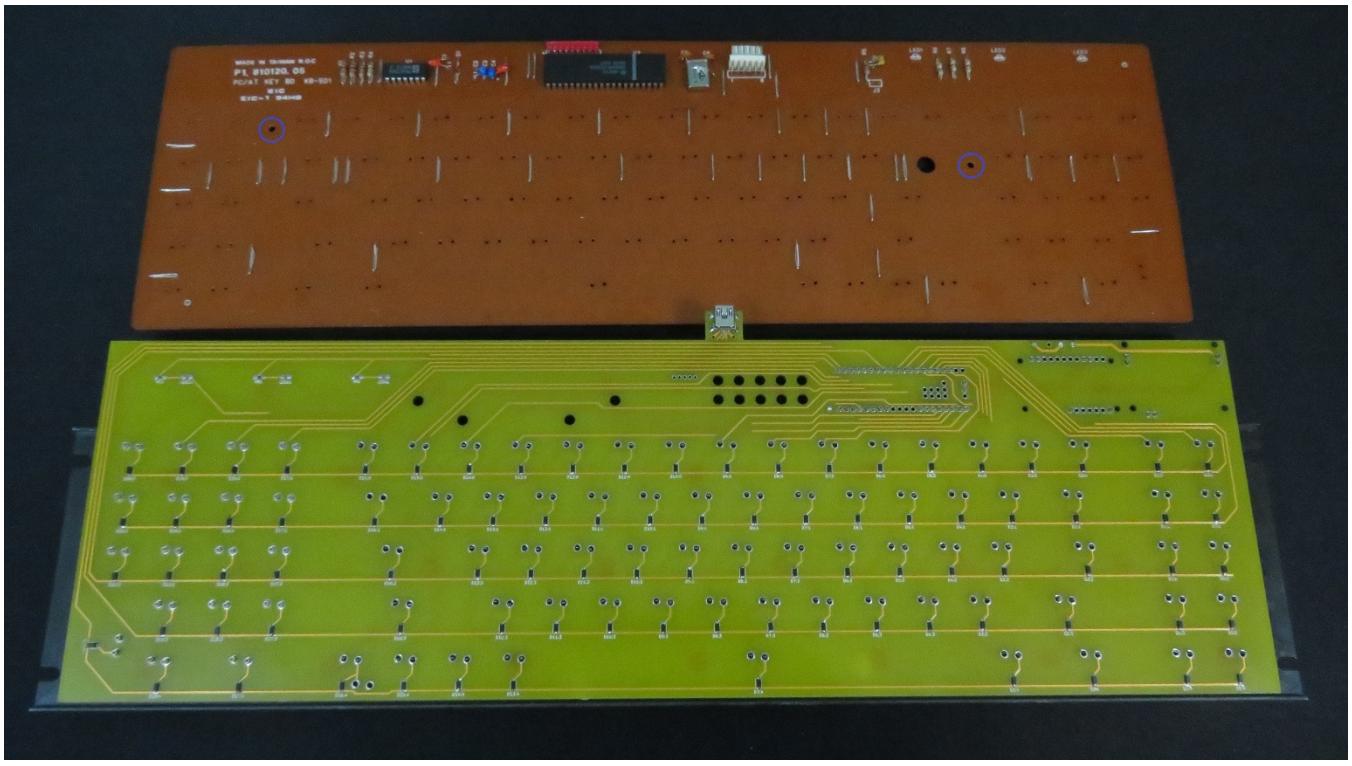


You can see how to solder applied to the through-holes would traditionally hold the switches in place.

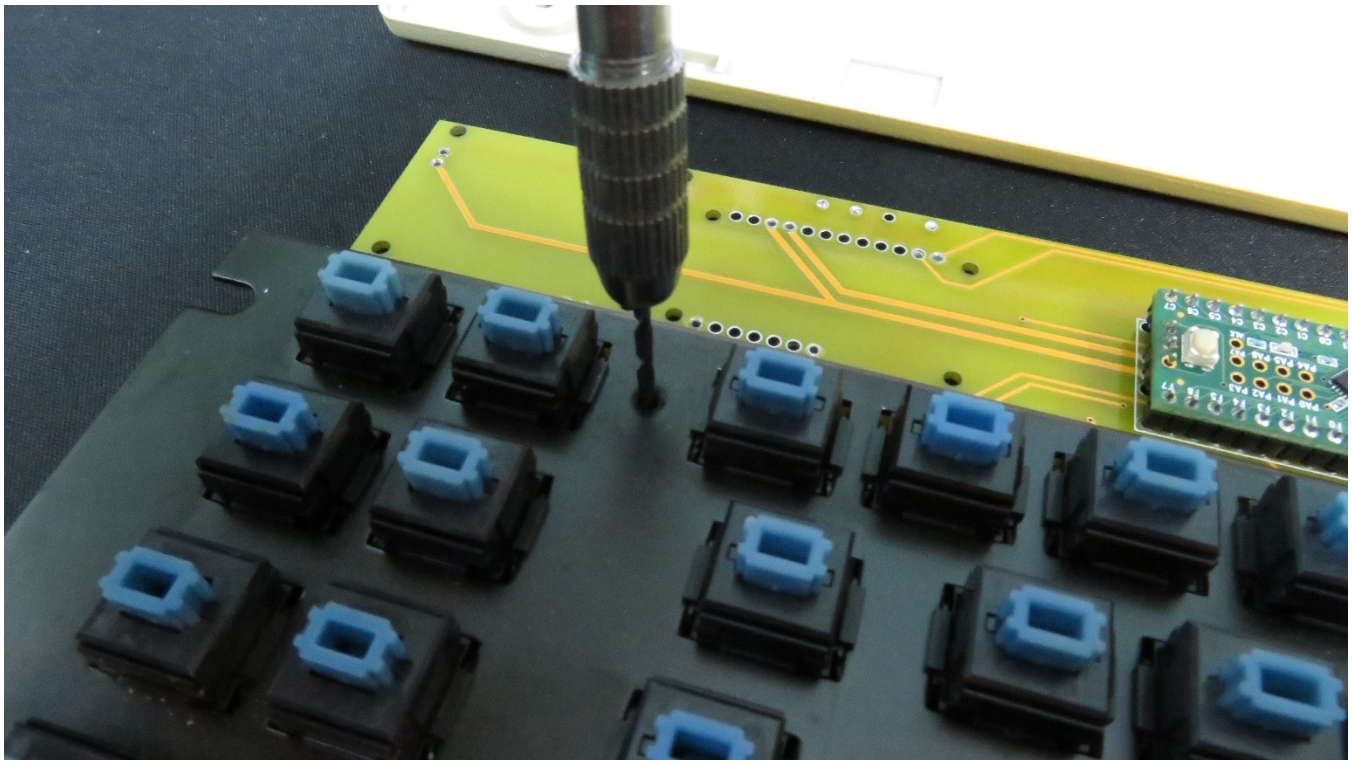


Desoldering complete, now the old PCB can be peeled away in one piece.

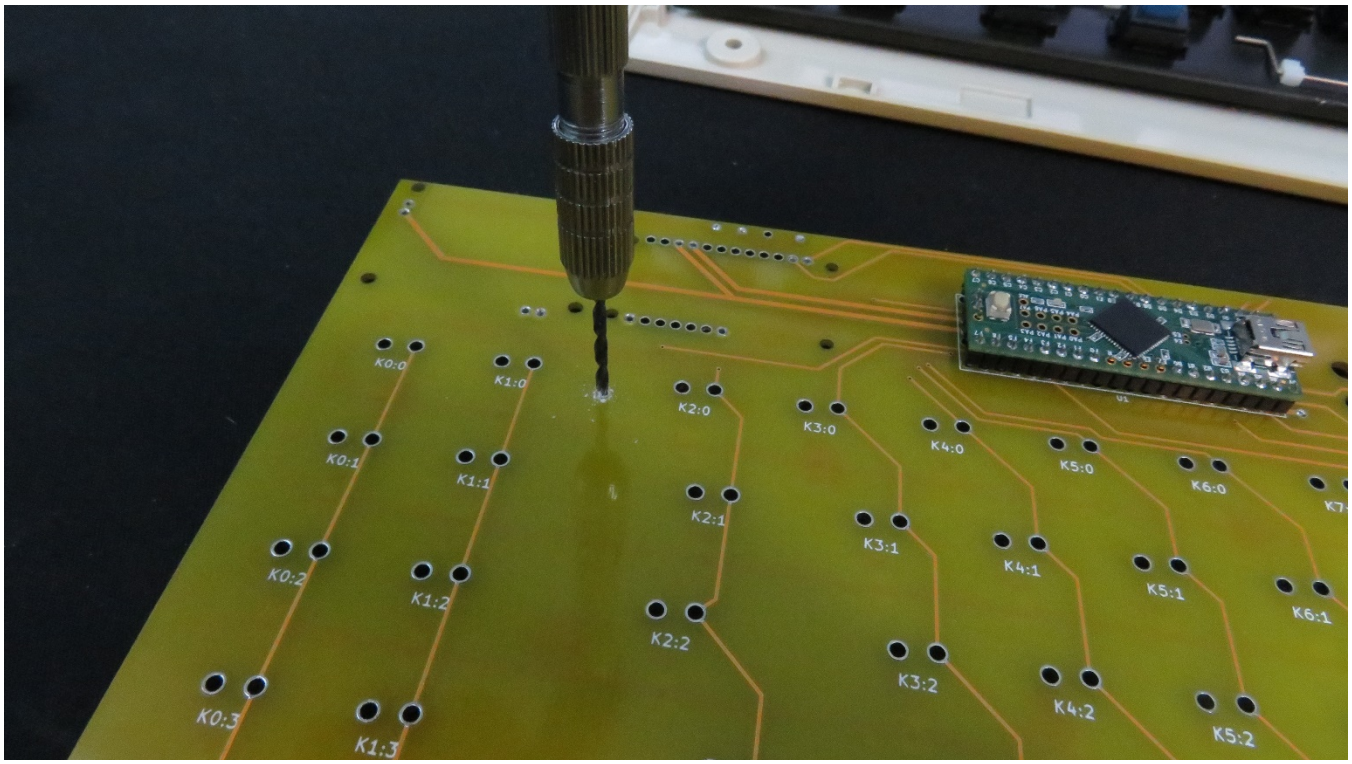
A single layer PCB was implemented as a cost saving method with small pieces of wire used to jump the traces etched on the other side.

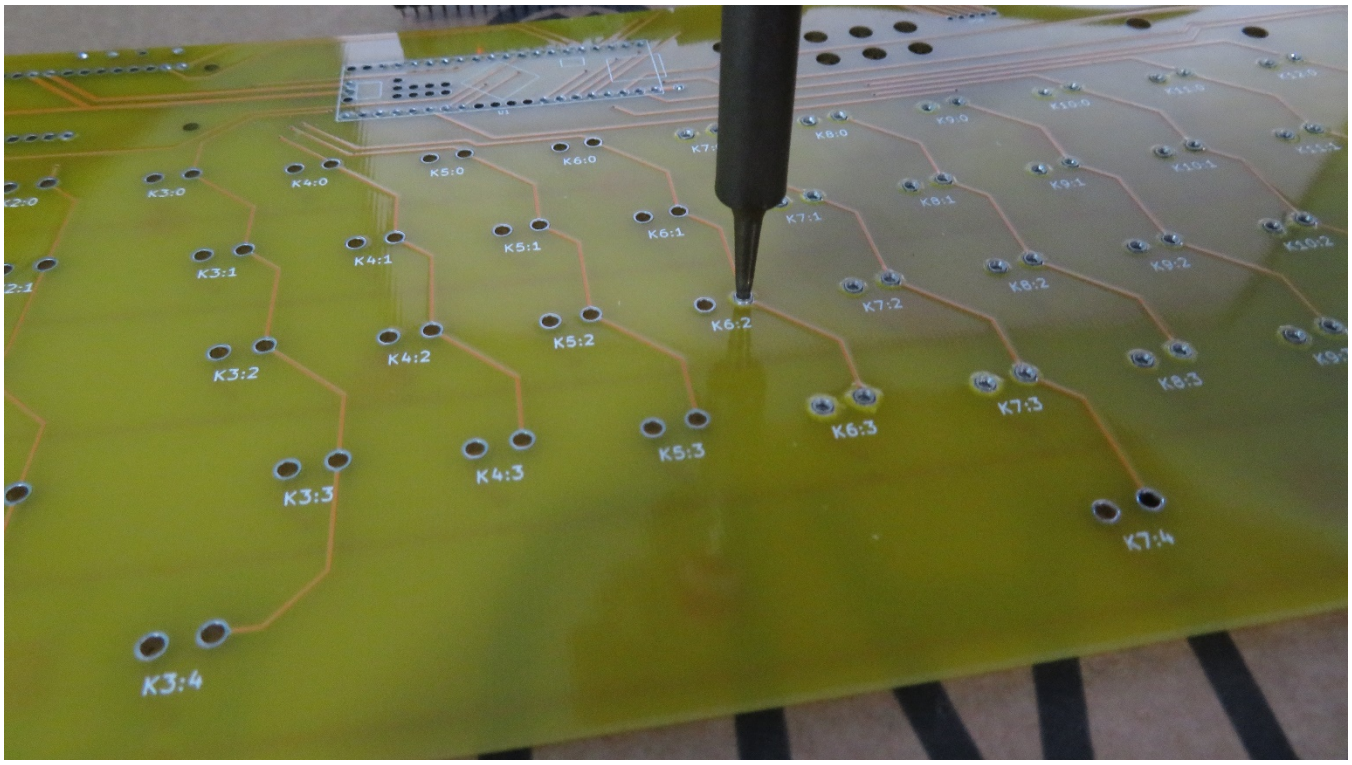


I did not notice the screws and (circled) subsequent holes further securing the plate to the PCB until I performed a test fit.



The screw hole was started with the hand drill and finished with the power drill. Luckily there were no traces running through these two locations.





Finishing up the holtite socket installation for the switches. To press them in, I have found that using a round, B-shape tip with the station set to 275º C in short periods can help any prevent pads from lifting.

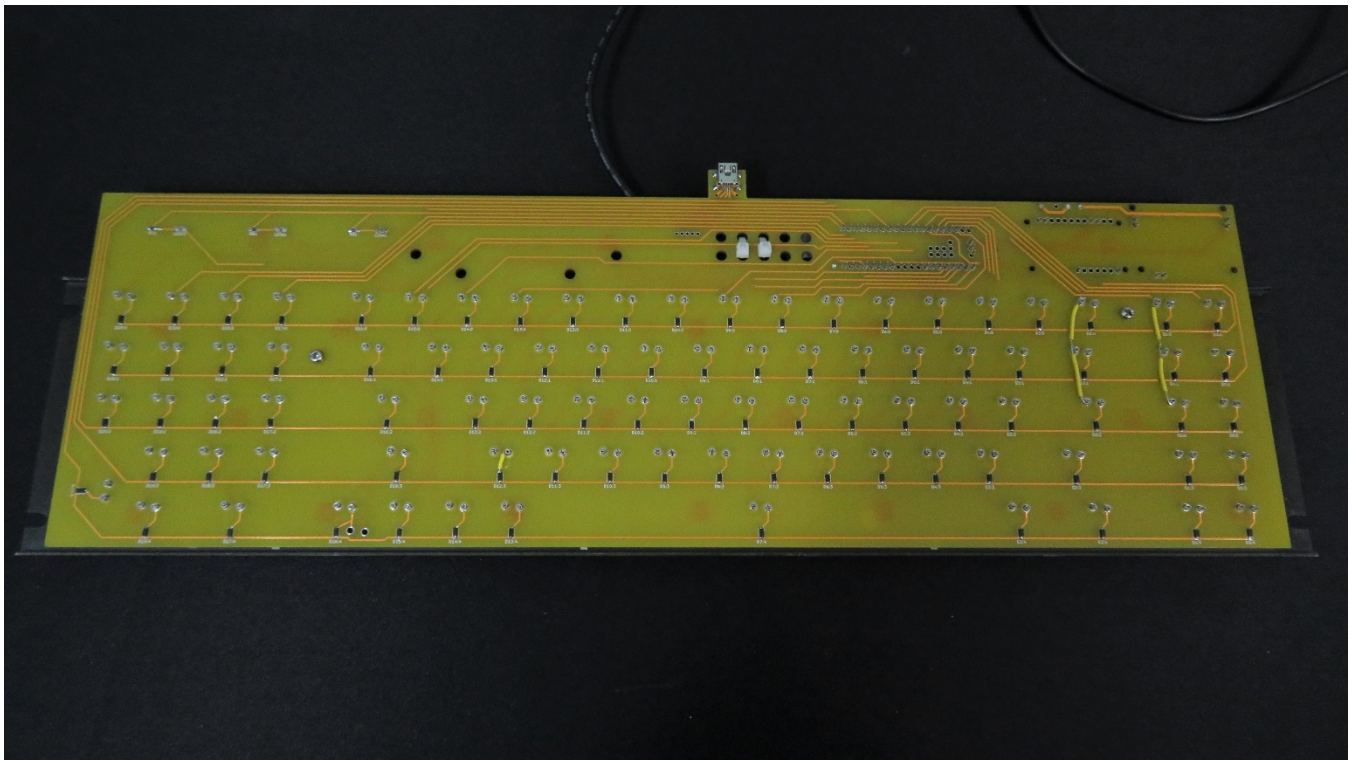
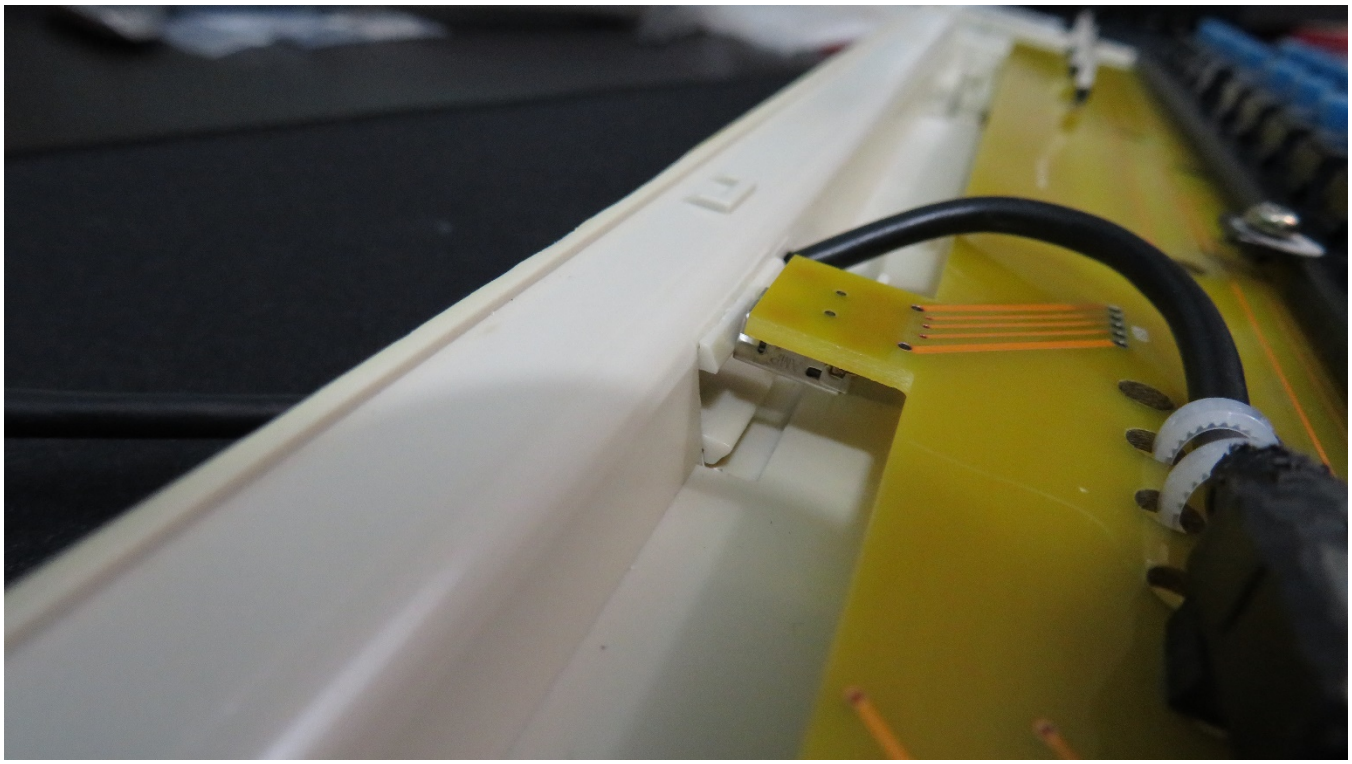
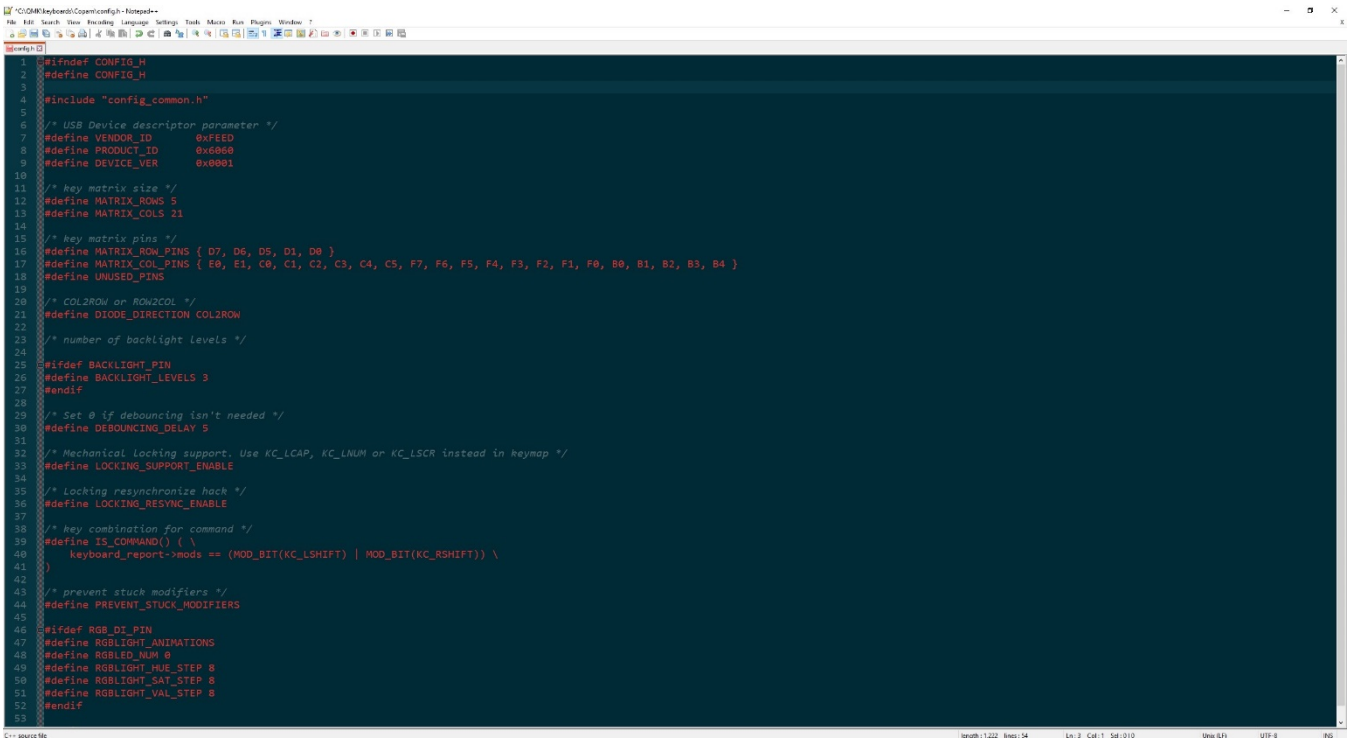


Plate screwed in and socket installation finished. Unfortunately, I lifted a few traces, but they were easily fixed by bridging the columns with some wire.

With the USB cable secured and filed down to clear to top housing, everything fits perfect. The original keyboard lock lights required too much power, so they were replaced with white 5x4x3 LEDs.

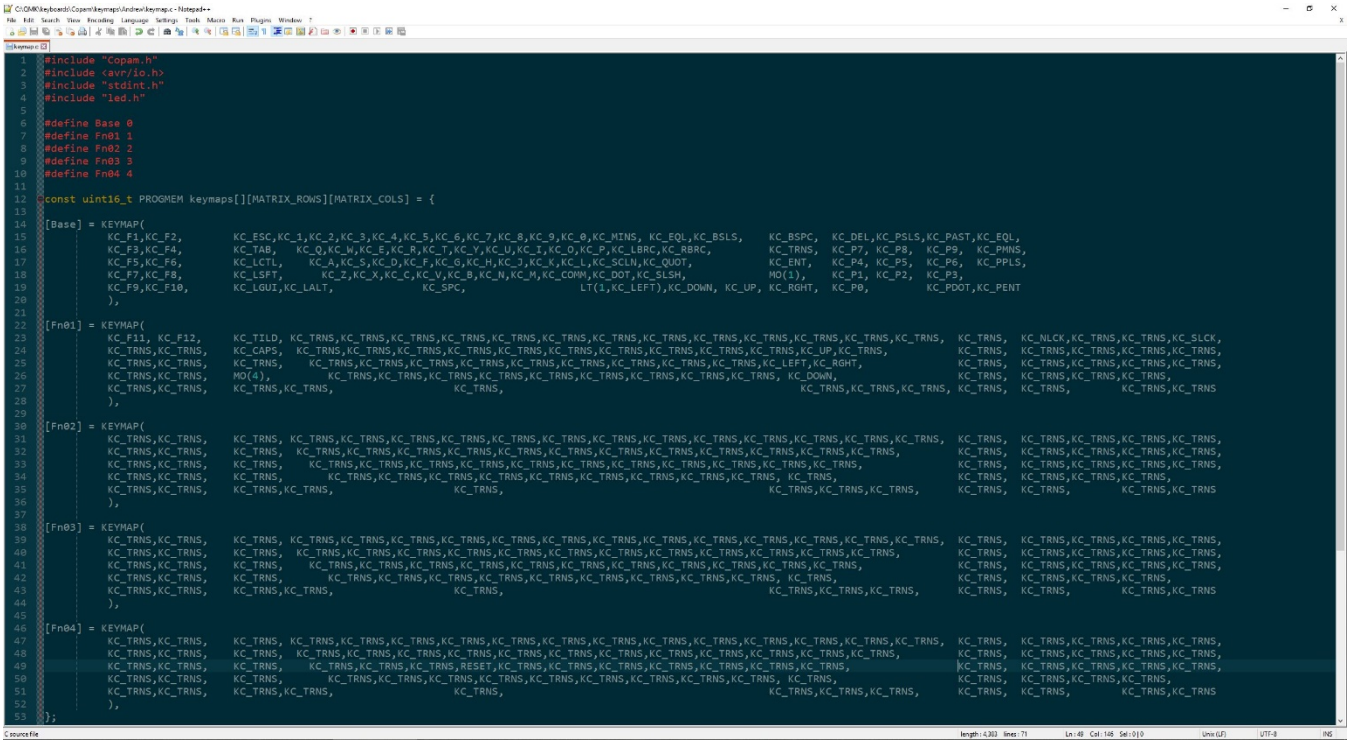


That USB connecter I didn't think would be implemented ended up fixing a small bother of mine. It fits tight enough against a broken trim piece to close off the large gap in the rear.



```
#ifndef CONFIG_H
#define CONFIG_H

#include "config_common.h"

/* USB Device descriptor parameter */
#define VENDOR_ID       0xFEED
#define PRODUCT_ID      0x6060
#define DEVICE_VER      0x0001

/* key matrix size */
#define MATRIX_ROWS 5
#define MATRIX_COLS 21

/* key matrix pins */
#define MATRIX_ROW_PINS { D7, D6, D5, D1, D0 }
#define MATRIX_COL_PINS { E0, E1, C0, C1, C2, C3, C4, C5, F7, F6, F5, F4, F3, F2, F1, F0, B0, B1, B2, B3, B4 }
#define UNUSED_PINS

/* COL2ROW or ROW2COL */
#define DIODE_DIRECTION COL2ROW

/* number of backlight levels */

#ifdef BACKLIGHT_PIN
#define BACKLIGHT_LEVELS 3
#endif

/* Set 0 if debouncing isn't needed */
#define DEBOUNCING_DELAY 5

/* Mechanical locking support. Use KC_LCAP, KC_LNUM or KC_LSCR instead in keymap */
#define LOCKING_SUPPORT_ENABLE

/* Locking resynchronize hack */
#define LOCKING_RESYNC_ENABLE

/* key combination for command */
#define IS_COMMAND() ( \
    keyboard_report->mods == (MOD_BIT(KC_LSHIFT) | MOD_BIT(KC_RSHIFT)) \
)

/* prevent stuck modifiers */
#define PREVENT_STUCK_MODIFIERS

#ifdef RGB_DI_PIN
#define RGBLIGHT_ANIMATIONS
#define RGBLED_NUM 0
#define RGBLIGHT_HUE_STEP 8
#define RGBLIGHT_SAT_STEP 8
#define RGBLIGHT_VAL_STEP 8
#endif
```

Building the firmware using an open-source project written in C, called QMK. Here we define the rows, columns, and their corresponding I/O pins on the Teensy. When a switch is pressed, this diode matrix will tell the chip where the input location is using coordinates of the two electrical signals.



```
#include "Copam.h"
#include <avr/io.h>
#include <stdint.h>
#include "led.h"

#define Base 0
#define Fn01 1
#define Fn02 2
#define Fn03 3
#define Fn04 4

const uint16_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {

[Base] = KEYMAP(
         KC_F1,KC_F2,        KC_ESC,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_BSLS,    KC_BSPC,  KC_DEL,KC_PSLS,KC_PAST,KC_EQL,
         KC_F3,KC_F4,        KC_TAB,   KC_Q,KC_W,KC_E,KC_R,KC_T,KC_Y,KC_U,KC_I,KC_O,KC_P,KC_LBRC,KC_RBRC,         KC_TRNS,  KC_P7, KC_P8,  KC_P9,  KC_PMNS,
         KC_F5,KC_F6,        KC_LCTL,    KC_A,KC_S,KC_D,KC_F,KC_G,KC_H,KC_J,KC_K,KC_L,KC_SCLN,KC_QUOT,          KC_ENT,   KC_P4, KC_P5,  KC_P6,  KC_PPLS,
         KC_F7,KC_F8,        KC_LSFT,      KC_Z,KC_X,KC_C,KC_V,KC_B,KC_N,KC_M,KC_COMM,KC_DOT,KC_SLSH,          MO(1),    KC_P1, KC_P2,  KC_P3,
         KC_F9,KC_F10,       KC_LGUI,KC_LALT,            KC_SPC,                 LT(1,KC_LEFT),KC_DOWN, KC_UP, KC_RGHT,  KC_P0,        KC_PDOT,KC_PENT
         ),

[Fn01] = KEYMAP(
         KC_F11, KC_F12,     KC_TILD, KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,  KC_TRNS,  KC_NLCK,KC_TRNS,KC_TRNS,KC_SLCK,
         KC_TRNS,KC_TRNS,    KC_CAPS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_UP,KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,    KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_LEFT,KC_RGHT,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    MO(4),        KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS, KC_DOWN,                KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,KC_TRNS,                  KC_TRNS,                                                KC_TRNS,KC_TRNS,KC_TRNS, KC_TRNS,  KC_TRNS,        KC_TRNS,KC_TRNS
         ),

[Fn02] = KEYMAP(
         KC_TRNS,KC_TRNS,    KC_TRNS, KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,  KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,    KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,      KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS, KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,KC_TRNS,                  KC_TRNS,                                                KC_TRNS,KC_TRNS,KC_TRNS, KC_TRNS,  KC_TRNS,        KC_TRNS,KC_TRNS
         ),

[Fn03] = KEYMAP(
         KC_TRNS,KC_TRNS,    KC_TRNS, KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,  KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,    KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,      KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS, KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,KC_TRNS,                  KC_TRNS,                                                KC_TRNS,KC_TRNS,KC_TRNS, KC_TRNS,  KC_TRNS,        KC_TRNS,KC_TRNS
         ),

[Fn04] = KEYMAP(
         KC_TRNS,KC_TRNS,    KC_TRNS, KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,  KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,    KC_TRNS,KC_TRNS,KC_TRNS,RESET,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,      KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS,KC_TRNS, KC_TRNS,          KC_TRNS,  KC_TRNS,KC_TRNS,KC_TRNS,
         KC_TRNS,KC_TRNS,    KC_TRNS,KC_TRNS,                  KC_TRNS,                                                KC_TRNS,KC_TRNS,KC_TRNS, KC_TRNS,  KC_TRNS,        KC_TRNS,KC_TRNS
         ),
};
```

Defining the keymap and function layers using the designated QMK keycodes. Even if the keyboard did work with my converter, each press would output the same code to the computer as is printed on the keycap. Remapping each key to a more normal layout is the biggest benefit of this retrofit.

```c
#include "led.h"
#include "Copam.h"

void led_init_ports(void) {
    DDRB |= (1<<5) | (1<<6) | (1<<7); // OUT
}

void led_set_kb(uint8_t usb_led) {
//    led_set_user(usb_led);
    if (usb_led & (1<<USB_LED_NUM_LOCK)) {
      // Turn num Lock on
       PORTB &= ~(1<<5);
    } else {
      // Turn numLock off
        PORTB |= (1<<5);
    }
     if (usb_led & (1<<USB_LED_SCROLL_LOCK)) {
      // Turn scroll Lock on
       PORTB |= (1<<6);
    } else {
      // Turn scroll Lock off
        PORTB &= ~(1<<6);
    }
     if (usb_led & (1<<USB_LED_CAPS_LOCK)) {
      // Turn caps Lock on
       PORTB |= (1<<7);
    } else {
      // Turn caps Lock off
        PORTB &= ~(1<<7);
    }

}
```

Writing the code for the lock LEDs was the most frustrating part of this entire process. Creating the switch matrix and keymap worked flawlessly on the first attempt flashing the hex file but took many more before the LEDs to behaved correctly.

It's alive! Each component now sits in place and functions as it should. The switch feel of Blue Alps can truly be appreciated now that the keyboard is operational. Big thanks to the community members that checked my PCB schematic! The most rewarding part about this was knowing I saved something that many others would have harvested the switches from then discarded.