#define HOST_OS 2 #include "../zmk-nodefree-config/helper.h" #include "../zmk-nodefree-config/international_chars/german.dtsi" #include <behaviors.dtsi> #include <dt-bindings/zmk/keys.h> #include <dt-bindings/zmk/bt.h> #include <dt-bindings/zmk/rgb.h> #include <dt-bindings/zmk/backlight.h> #include <dt-bindings/zmk/ext_power.h>

#define _____ &trans

// #define DEF 0 // layer shortcuts, must match order in which they are defined below // #define NAV 1 // #define NUM 2 // #define GER 3

// tap: sticky shift | double tap: capsword ZMK_BEHAVIOR(ss_cw, tap_dance, tapping-term-ms = <200>; bindings = <&sk LSHFT>, <&caps_word>;)

// tap: backspace | shift + tap: delete | hold: num layer // ZMK_BEHAVIOR(bs_del_num, mod_morph, // bindings = <&lt NUM BSPC>, <&kp DEL>; // mods = <(MOD_LSFT|MOD_RSFT)>; //)

// euro sign ZMK_UNICODE_SINGLE(euro_sign, N2, N0, A, C) // €

// replace a/o/u/s with German umlauts when NAV and NUM are held jointly // ZMK_CONDITIONAL_LAYER(NAV NUM, GER)

// combos #undef COMBO_TERM #define COMBO_TERM 40 // timeout of 40ms (default is 30ms if omitted) ZMK_COMBO(combo_copy, &kp LC(C), LB2 LB3, ALL) // Ctrl + C, active on all layers ZMK_COMBO(combo_paste, &kp LC(V), LB1 LB2, ALL) // Ctrl + V, active on all layers

/ { keymap { compatible = "zmk,keymap"; default_layer { // ———————————————————————————— // | ESC | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F9 | F9 | F10 | F11 | F12 | PSCRN | INS | DEL | // | ' | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | - | = | BKSP | HOME | // | TAB | Q | W | E | R | T | Y | U | I | O | P | [|] | | PG UP | // | CAPS | A | S | D | F | G | H | J | K | L | ; | ' | ENTER | PG DN | // | SHIFT | Z | X | C | V | B | N | M | , | . | / | SHIFT | UP | END | // | CTL | WIN | ALT | SPACE | ALT | 1 | CTRL | LEFT | DOWN | RIGHT | // ———————————————————————————— bindings = < &kp ESC &kp C_BRIGHTNESS_DEC &kp C_BRIGHTNESS_INC &rgb_ug RGB_TOG &rgb_ug RGB_EFF &bl BL_DEC &bl BL_INC &kp C_PREV &kp C_PP &kp C_NEXT &none &none &kp HASH &kp C_AL_LOCK &kp C_MUTE &kp C_MUTE &kp GRAVE &kp N1 &kp N2 &kp N3 &kp N4 &kp N5 &kp N6 &kp N7 &kp N8 &kp N9 &kp N0 &kp de_eszett &kp EQUAL &kp BSPC &mt LCMD C &kp Q &kp W &kp E &kp R &kp T &kp Z &kp U &kp I &kp O &kp P &kp de_ue &kp RBKT & &mt LCMD V RC(3,0) RC(3,1) RC(3,2) RC(3,3) RC(3,4) RC(3,5) RC(3,6) RC(3,7) RC(3,8) RC(3,9) RC(3,10) RC(3,11) RC(3,13) RC(3,15) &mo 1 &kp A &kp S &kp D &kp F &kp G &kp H &kp J &kp K &kp L &kp de_oe &kp de_ae &kp RET &kp PG_UP RC(4,0) RC(4,2) RC(4,3) RC(4,4) RC(4,5) RC(4,6) RC(4,7) RC(4,8) RC(4,9) RC(4,10) RC(4,11) RC(4,12) RC(4,14) RC(4,15) &kp LSHFT &kp LESS_THAN &kp Y &kp X

&kp C &kp V &kp B &kp N &kp M &kp COMMA &kp DOT &kp FSLH &kp RSHFT &kp RSHFT &kp UP &kp PG_DN RC(5,0) RC(5,1) RC(5,2) RC(5,6) RC(5,10) RC(5,11) RC(5,12) RC(5,13) RC(5,14) RC(5,15) &kp LCTRL &kp LGUI &kp LALT &kp SPACE &kp RCMD &kp RALT &kp RCTRL &kp LEFT &kp DOWN &kp RIGHT >; sensor-bindings = <&inc_dec_kp C_VOL_UP C_VOL_DN &inc_dec_kp C_VOL_UP C_VOL_DN>; }; raise { bindings = < &trans &rgb_ug RGB_BRD &rgb_ug RGB_BRI &rgb_ug RGB_SPD &rgb_ug RGB_SPI &rgb_ug RGB_EFF &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &kp ESC &kp F1 &kp F2 &kp F3 &kp F4 &kp F5 &kp F6 &kp F7 &kp F8 &kp F9 &kp F10 &kp F11 &kp F12 &trans &trans &trans &trans &trans &kp UP &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &bootloader &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &bt BT_PRV &bt BT_NX uiT &trans &trans &trans &trans &trans &trans &trans &trans >; sensor-bindings = <&inc_dec_kp C_VOL_UP C_VOL_DN &inc_dec_kp PG_UP PG_DN>; }; }; };