Keyboard Firmware

All the code.

- QMK Basics

QMK Basics

A short guide by PyroL. Free yourself from those prebaked, out-of-date and several-hundred-commits-behind tools!

All info available at https://docs.qmk.fm/

Getting QMK

Using your favorite command-line, clone the main repo by running:

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

If you have a fork, clone the main repo anyways, then run:

`git remote add upstream [your repo link]`, and then you can push to your own repo and pull from origin, aka the main repo.

Customizing QMK

In QMK, a keyboard's firmware is broken up into a few important files:

- **config.h** defines physical and electrical properties about the board: what it's recognized as on the computer, what microcontroller pins are connected to which rows and columns, and a bit of other QMK magic can go on in this file.
- **[keyboard name].h** defines the physical layout of the keyboard, so that the keymap arrays can send the right keycodes to the right key events.
- You mostly don't need to touch **rules.mk** or **[keyboard name].c**
- **keymaps/[keymap name]/keymap.c** is where most of the editing will happen.

```
/* Example: Clueboard default base layer */
[_BL] = LAYOUT(
  F(0),   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_GRV,  KC_BSPC,        KC_PGUP, \
  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_BSLS,             KC_PGDN, \
  KC_CAPS, KC_A,   KC_S,  KC_D,  KC_F,  KC_G,  KC_H,  KC_J,  KC_K,  KC_L,   KC_SCLN, KC_QUOT,
KC_NUHS,  KC_ENT,                  \
  KC_LSFT, KC_NUBS, KC_Z,  KC_X,  KC_C,  KC_V,  KC_B,  KC_N,  KC_M,  KC_COMM, KC_DOT,  KC_SLSH,
  KC_RO,   KC_RSFT,        KC_UP,         \
```

```
    KC_LCTL, KC_LGUI, KC_LALT, KC_MHEN,        KC_SPC,KC_SPC,                KC_HENK,  KC_RALT,  KC_RCTL,
    MO(_FL), KC_LEFT, KC_DOWN, KC_RGHT),
```

QMK layouts are just C arrays QMK compares to the header file that define physical properties of the board to determine what gets sent to the computer when certain key events are detected. Most keycodes are baked in to QMK, and a full list is available at https://docs.qmk.fm/#/keycodes. There are lots of exotic and surprisingly flexible and powerful keycodes that you may find useful, so I encourage you to read up!

To make the data easier to parse, most keymap files have a line break for every row on the keyboard, so you can easily edit a specific key on your keyboard without having to count it out. From there, just change the keycode to whatever you want it to be. For example, if I wanted to make Caps Lock my control key on the above layer, I would change `KC_CAPS` on the 3rd row from the top to `KC_LCTRL`, then reflash.

Layers