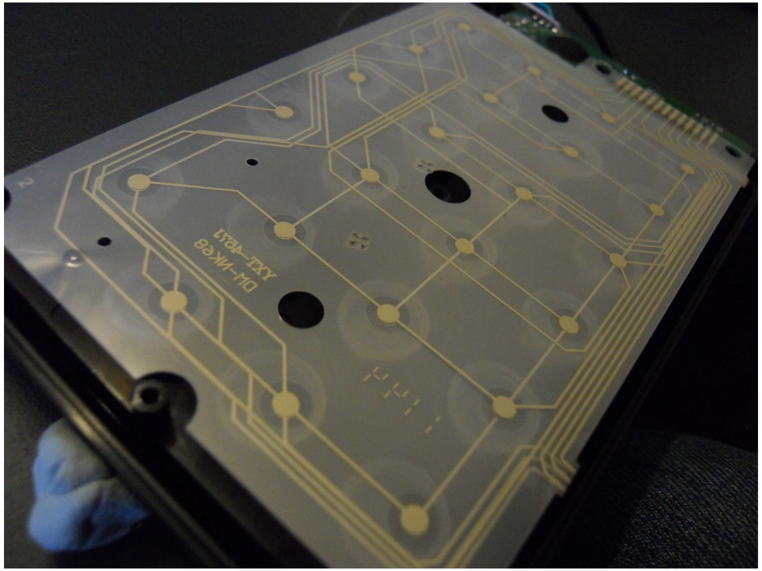
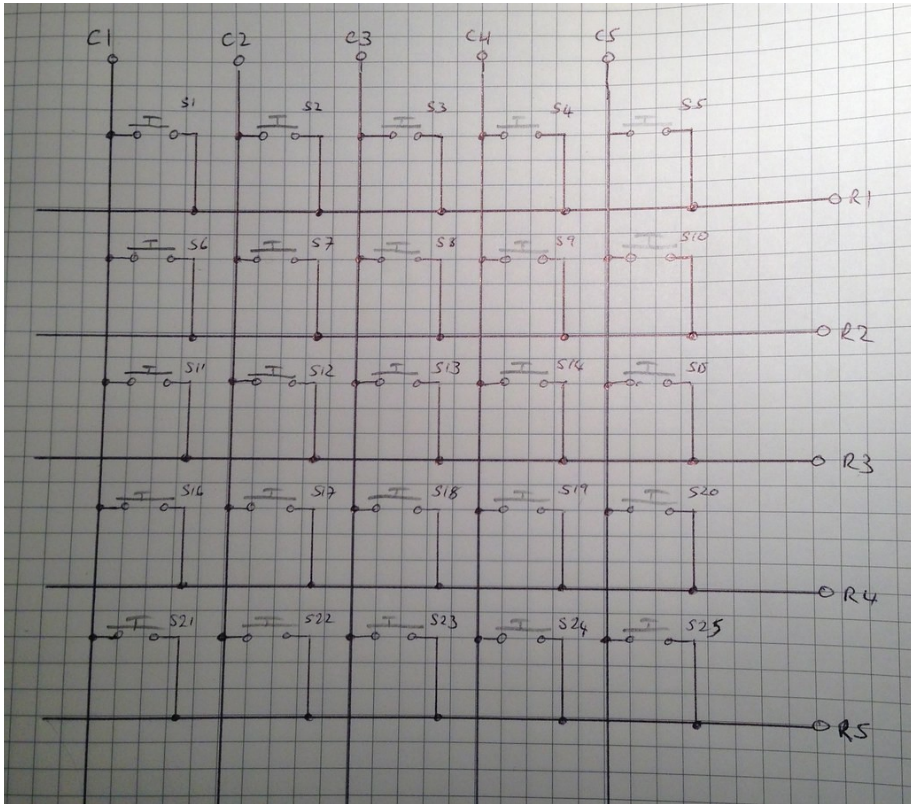
QMK: Customizing your Keyboard Firmware

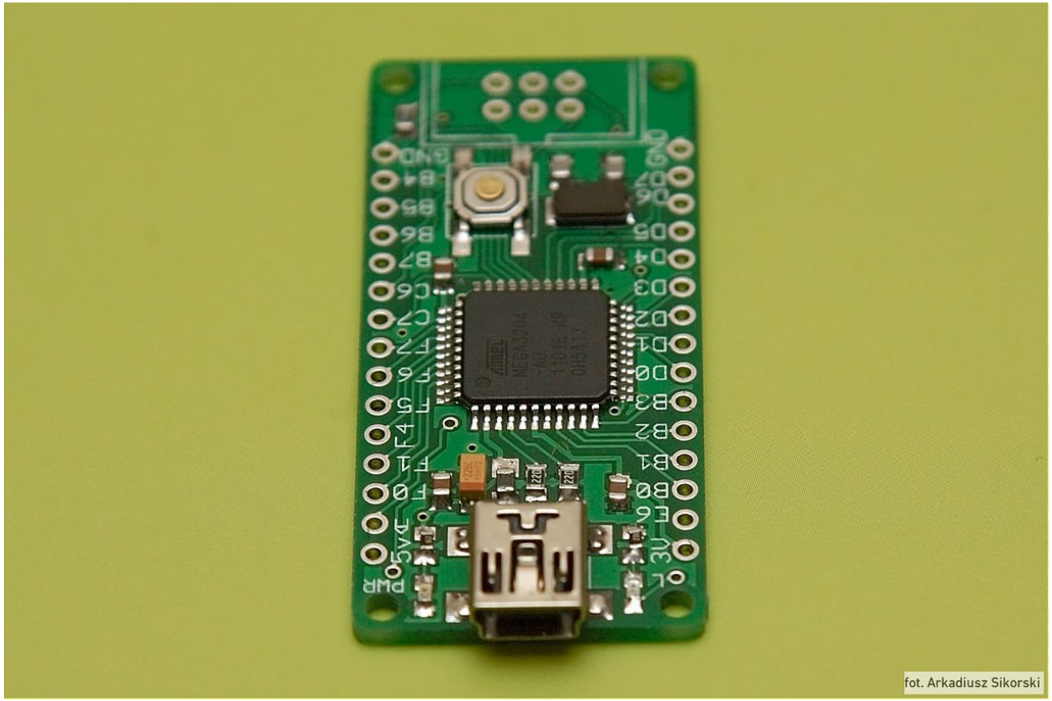
Why and how to reprogram mechanical keyboard behavior using Quantum Mechanical Keyboard

Mechanical Or Not



How Does a Keyboard Work?





fot. Arkadiusz Sikorski

Getting QMK

- https://qmk.fm/
- https://github.com/qmk/qmk_firmware
- https://docs.qmk.fm/#/
- https://config.qmk.fm/#/
- python3 -m pip install --user qmk
- sudo pacman -S qmk

Setting up QMK

- qmk setup
- qmk setup <github_username>/qmk_firmware
- Ubuntu
  - echo 'PATH="$HOME/.local/bin:$PATH"' >> $HOME/.bashrc && source $HOME/.bashrc
  - Adds $HOME/.local/bin back to PATH

Applying Firmware

- qmk compile -kb <keyboard> -km <keymap>
  - Keyboard is <manufacturer>/<board>/<version>
    - Ex: keebio/quefrency/rev2
    - Same order as directories
  - “default” keymap is… the default.
- qmk flash -kb <keyboard> -km <keymap>

Making a Keymap

- qmk new-keymap -kb <keyboard_name>
- Three files:
  - config.h
  - keymap.c
  - rules.mk
- Precedence: keymap dir > version dir > keyboard dir
- https://docs.qmk.fm/#/keycodes

Configuration

- qmk config
  - <subcommand|general|default>[.<key>][=<value>]
  - Ex: qmk config default.keymap=clicky
- User options:
  - user.keyboard
  - user.keymap
  - user.name

Command Documentation

- https://docs.qmk.fm/#/cli_commands
- Use man

Vim-Like Features

- Dynamic Macros
  - DYNAMIC_MACRO_ENABLE
  - 2 macros (DM_REC1, DM_PLY1, DM_REC2, DM_PLY2)
  - Cannot (by default) be repeated x times
  - DM_RSTP or pressing the start key again finishes the macro recording
  - Recursive macros are not your friend!
- Leader Key
  - LEADER_ENABLE
  - Uses matrix_scan_user

N-key Rollover

- #define FORCE_NKRO
- Default without NKRO is 6 keys at once
- With NKRO, 248 keys at once

Key Lock

- KEY_LOCK_ENABLE
- Keycode: KC_LOCK
- Next key after pressing key lock stays held down until key lock is pressed again

Mod-Tap

- MT(modifier key, key)
  - Have to use MOD_ instead of KC_ prefix for the modifier key
- Is the modifier key when held
- Is the other key when tapped

Auto Shift

- AUTO_SHIFT_ENABLE
- Does not work with modifiers by default
- Tap a key slightly longer to get it shifted
- The timing takes practice

One Shot Keys

- #define ONESHOT_TAP_TOGGLE x
  - x is number of presses to lock the key
- #define ONESHOT_TIMEOUT y
  - Y is milliseconds until the oneshot key is released
- Keeps the key active until the next key is pressed
- Tapping multiple times locks the key
- OSM(mod), uses MOD_ codes

Mouse Keys

- MOUSEKEY_ENABLE
- Accelerated, kinetic, constant, or combined
- Press mouse buttons 1-8
- Move cursor
- Move wheel in the four directions

Combos

- COMBO_ENABLE
- #define COMBO_COUNT x
  - x is the number of combos defined
- In keymap give an enum, define a combo, and use a combo function
  - Also can use process_combo_event

Tap Dance

- TAP_DANCE_ENABLE
- #define Tapping_Term x
  - x is the timeout between taps
- Do different things when a key is repeatedly tapped
- TD(dance) is used instead of a keycode
- Defined in enum, then tap_dance_actions
- Different types of dance available for tap_dance_actions

Layers

- Overlay new sets of key mappings
- DF(lay) makes lay the new default layer
- MD(lay) activates a layer when held
- LM(lay, mod) is MD with the modifier active
- LT(lay, kc) MD when held, sends key on tap
- OSL(lay) activates layer until next keypress
- TG(lay) toggles layer
- TO(lay) activates the layer and deactivates all others
- TT(lay) tap-toggle for a layer

Macros

- process_record_user / post_process_record_user
- SEND_STRING() method sends a string
- SS_TAP() to press and release a key
- SS_DOWN() to press a key
- SS_UP() to release a key
- send_string() works for strings in memory
- Even more complex execution possible

Stenography

- Plover by Open Steno Project
  - “plover” from AUR
  - AppImage
  - https://github.com/openstenoproject/plover
- Needs NKRO
- Switch between a regular keyboard and steno
- STENO_ENABLE
  - Uses 3 USB endpoints (may not work with mouse emulation)
- #include keymap_steno.h
- In matrix_init_user(), steno_set_mode
  - STENO_MODE_GEMINI
  - STENO_MODE_BOLT

Lighting

- BACKLIGHT_ENABLE
- LED_MATRIX_ENABLE
- LED_MATRIX_DRIVER
- RGBLIGHT_ENABLE
  - https://docs.qmk.fm/#/feature_rgblight
- RGB_MATRIX_ENABLE

Rarer Features

- Audio
- Bluetooth
- Encoders
- Split Keyboard
  - SPLIT_KEYBOARD

Space Saving Tricks

- Keep an eye on compilation memory %
- Disable features you don’t use
- Enable Link Time Optimization
  - LTO_ENABLE = yes (in rules.mk)
  - Disables old TMK Macros and Functions, does not disable QMK Macros and Layers
- See https://beta.docs.qmk.fm/developing-qmk/qmk-reference/config_options
- See https://thomasbaart.nl/2018/12/01/reducing-firmware-size-in-qmk/ (slightly out of date)

Bonus: Weirdly Specific Features

- Grave Escape: share the grave key with escape
- Space Cadet Shift: tapped shift keys give ( )
- Swap-Hands: key that mirrors the keyboard
- A terminal
- Unicode
- WPM rolling average
- MIDI
- Raw HID
- Bootmagic

- Custom Matrix
- DIP Switches
- Haptic Feedback
- Joystick
- LED Indicators
- PS/2 Mouse
- Thermal Printer
- Velocikey

Image Credits

- Slide 2
  - "Cherry MX Switch" by SparkFunElectronics is licensed with CC BY 2.0.
  - "The keyboard membrane" by makototakeuchi is licensed with CC BY 2.0.
- Slide 3
  - "A 5x5 simple keyboard matrix" by pdcawley is licensed with CC BY-NC-SA 2.0.
  - "Arduino Leonardo Nano Pro Mini ATmega32U4 comp. / Photo by Arkadiusz Sikorski" by Arkadiusz Sikorski vel ArakuS is licensed with CC BY-NC 2.0.