

Custom Keyboards

Building your own Layouts, Firmware and Hardware

Carsten Thiel

October 23, 2026

Keyboards

What you're probably thinking of

Keyboards

What you're probably thinking of



Cherry Keyboard

Keyboards

What you're probably thinking of



Cherry Keyboard



Apple Magic Keyboard

Keyboards

What you're probably thinking of



Cherry Keyboard



Apple Magic Keyboard



Nuphy Air 75v3

About me

About me

\$ whoami

- ▶ Carsten Thiel
- ▶ Platform developer @ Sparebanken Norge
- ▶ Background in Mathematics
- ▶ Born in Germany – living in Bergen
- ▶ Speak and use several languages

Layouts

Some layout examples

Some layout examples



Apple Magic Keyboards US Intl and Norwegian

Some layout examples



Apple Magic Keyboards US Intl and Norwegian — Keychron Nordic

How does this work?

How does this work?

- ▶ Keyboards send 8-bit keycodes
- ▶ Layout is an OS setting!

XKB layouts

XKB layouts

X Keyboard Extension

XKB layouts

X Keyboard Extension

- ▶ De-facto Linux standard

XKB layouts

X Keyboard Extension

- ▶ De-facto Linux standard
- ▶ X11 – works in Wayland too

XKB basics

XKB basics

English/American

```
key <AC07> { [ j, J ] };
key <AC08> { [ k, K ] };
key <AC09> { [ l, L ] };
key <AC10> { [ semicolon, colon ] };
key <AC11> { [ apostrophe, quotedbl ] };
```

XKB basics

English/American

```
key <AC07> { [ j, J ] };
key <AC08> { [ k, K ] };
key <AC09> { [ l, L ] };
key <AC10> { [ semicolon, colon ] };
key <AC11> { [ apostrophe, quotedbl ] };
```

Norwegian

```
key <AC07> { [ j, J ] };
key <AC08> { [ k, K ] };
key <AC09> { [ l, L ] };
key <AC10> { [ oslash, Oblique ] };
key <AC11> { [ ae, AE ] };
```

* Simplified

XKB Norwegian definition

XKB Norwegian definition

```
default partial alphanumeric_keys
xkb_symbols "basic" {

    include "latin(type2)"

    name[Group1]="Norwegian";

    key <AB09> { [      period,      colon,      ellipsis,      periodcentered ] };
    key <AB10> { [      minus,      underscore,      endash,      emdash ] };
    key <AD04> { [      r,      R,      registered,      trademark ] };
    key <AD10> { [      p,      P,      Greek_pi,      Greek_PI ] };
    key <AE05> { [      5,      percent,      onefhalf,      0x1002030 ] };

    key <AE11> { [      plus,      question,      plusminus,      questiondown ] };
    key <AE12> { [      backslash,      dead_grave,      dead_acute,      notsign ] };

    key <AC10> { [      oslash,      Ooblique,      dead_acute,      dead_doubleacute ] };
    key <AC11> { [      ae,      AE,      dead_circumflex,      dead_caron ] };
    key <TLDE> { [      bar,      section,      brokenbar,      paragraph ] };

    key <BKSL> { [      apostrophe,      asterisk,      dead_doubleacute,      multiply ] };
    key <LSGT> { [      less,      greater,      onefhalf,      threequarters ] };

    include "nbsp(level3n)"
    include "keypad(ossmath)"
    include "kpdl(comma)"

    include "level3(ralt_switch)"

};
```

Modifiers

Modifiers

► Shift

Modifiers

- ▶ Shift
- ▶ Alt & AltGr / Option

Modifiers

- ▶ Shift
- ▶ Alt & AltGr / Option
- ▶ Control

Modifiers

- ▶ Shift
- ▶ Alt & AltGr / Option
- ▶ Control
- ▶ Super / Command / Windows

Modifiers

- ▶ Shift
- ▶ Alt & AltGr / Option
- ▶ Control
- ▶ Super / Command / Windows

⇒ Additional uses for keys

Modify and make your own?

Modify and make your own?

- ▶ Ultimate flexibility
- ▶ Combine your favourite languages!
- ▶ Maybe there is none that meets your needs?

My custom layout us_cat

My custom layout us_cat

~ 1	Š 2	@ 3	Ž 3	# 3	Č 4	\$ 4	Ć 5	% 5	Đ 6	^ 6	^ 7	& 7	ˆ 8	* 8	ˆ 9	(9	{ [	)]	}]	- -	- -	+ =	^ '	Backspace	
Tab		Q £	Å £	W £	E €	¢ €	R ®	T ™	Y ¥	U ¥	Ü ¥	I ø	Ø ø	O ö	Ö ö	P å	Å å	{ [	` "	' "	}]	' "	 \		
		A Ä	Ä Ä	S ß	ß ß	D §	F §	G °	H ◀	J ↓	↓ ↓	K ↑	↑ ↑	L ⇒	⇒ →	:	` "	" "	' "	' "	' "	' "	Enter		
Shift		Z Æ	Æ Æ	X ≈	≈ ×	C ©	© ©	V ...	B ∞	N ¬	¬ ¬	M μ	μ μ	< <	< «	> >	> »	? /	, "	, "	, "	, "	Shift		
Ctrl	⌘	Alt													Alt Gr	⌘		Ctrl							

https://github.com/schildwaechter/us_cat

But why are letters arranged like this?

But why are letters arranged like this?



Remington No. 2 from 1878, CC-BY-SA 4.0 Martin Howard, Wikimedia Commons

So – 150 years of Qwerty?

So – 150 years of Qwerty?

Pros and Cons

- ▶ It's **the** standard
- ▶ Left-handed focus
- ▶ Lots of finger movement/stretching
- ▶ *Legacy*

So – 150 years of Qwerty?

Pros and Cons

- ▶ It's **the** standard
- ▶ Left-handed focus
- ▶ Lots of finger movement/stretching
- ▶ *Legacy*

Alternatives

- ▶ Dvorak by August Dvorak, 1936
- ▶ Colemak by Shai Coleman, 2006

So – 150 years of Qwerty?

Pros and Cons

- ▶ It's **the** standard
- ▶ Left-handed focus
- ▶ Lots of finger movement/stretching
- ▶ *Legacy*

Alternatives

- ▶ Dvorak by August Dvorak, 1936
- ▶ Colemak by Shai Coleman, 2006
(and many more)

Dvorak and Colemak

Dvorak and Colemak

Pros

- ▶ Optimised for finger movement
- ▶ Supported on all operating systems

Dvorak and Colemak

Pros

- ▶ Optimised for finger movement
- ▶ Supported on all operating systems

Cons

- ▶ Non-standard and not in hardware
- ▶ Lots of variants and successors

Colemak?

Colemak - 1000 most popular
words in English



QWERTY - 1000 most popular
words in English



TypingDoneWell.com



<https://typingdonewell.com/blog/what-is-colemak-layout-and-is-it-good-with-my-own-tests/>

Layout editing resources

Linux `xkb - man xkeyboard-config`

Plain text definition files in `/usr/share/X11/xkb`

Layout editing resources

Linux `xkb - man xkeyboard-config`

Plain text definition files in `/usr/share/X11/xkb`

Windows Microsoft Keyboard Layout Creator (MSKLC)

Visual editor and compiler using plain text sources

macOS Ukelele by SIL

Visual editor for native `.keymap` XML files

Layout editing resources

Linux `xkb - man xkeyboard-config`

Plain text definition files in `/usr/share/X11/xkb`

Windows Microsoft Keyboard Layout Creator (MSKLC)

Visual editor and compiler using plain text sources

macOS Ukelele by SIL

Visual editor for native `.keymap` XML files

Cross-Platform Keyman (MIT) by SIL (untested)

Software that runtime-intercepts keypresses

Now ...



Layers

Fn-key

Fn-key

- ▶ Not a modifier like before
- ▶ Exists only in the keyboard

⇒ Additional uses for keys

Programmable keyboards

Programmable keyboards

VIA/Vial Firmware extensions

Programmable keyboards

VIA/Vial Firmware extensions

- ▶ Customizable keymap
- ▶ USB/WebHID API

Programmable keyboards

VIA/Vial Firmware extensions

- ▶ Customizable keymap
- ▶ USB/WebHID API

QMK/ZMK Firmware (GPL/MIT)

Programmable keyboards

VIA/Vial Firmware extensions

- ▶ Customizable keymap
- ▶ USB/WebHID API

QMK/ZMK Firmware (GPL/MIT)

- ▶ Fully programmable
- ▶ “Infinite” complexity

Programmable keyboards

VIA/Vial Firmware extensions

- ▶ Customizable keymap
- ▶ USB/WebHID API

QMK/ZMK Firmware (GPL/MIT)

- ▶ Fully programmable
- ▶ “Infinite” complexity

- ▶ Proprietary solutions

Programmable keyboards

VIA/Vial Firmware extensions

- ▶ Customizable keymap
- ▶ USB/WebHID API

QMK/ZMK Firmware (GPL/MIT)

- ▶ Fully programmable
- ▶ “Infinite” complexity

- ▶ Proprietary solutions

⇒ Even more flexible and all in hardware

More functionality

More functionality

Features

- ▶ Advanced Configuration and Layers
- ▶ State and automagic
- ▶ RGB Lighting and Displays

More functionality

Features

- ▶ Advanced Configuration and Layers
- ▶ State and automagic
- ▶ RGB Lighting and Displays

Programming

- ▶ Home Row Mods
- ▶ Tap Dancing
- ▶ Macros

Ergonomics

Touch Typing

Touch Typing



Key-Finger mapping, CC-BY-SA Artur Mihno, Wikimedia Commons

Touch Typing

Trade-offs

- ▶ Fast, once mastered
- ▶ Not ideal on shoulders
- ▶ Can cause RSI



Key-Finger mapping, CC-BY-SA Artur Mihno, Wikimedia Commons

Split columnar

Split columnar



Keeb.io Iris LM

Custom keyboards

Custom keyboards

Design your own

Custom keyboards

Design your own

Physical layout

- ▶ Row-staggered
- ▶ Ortholinear
- ▶ Column-staggered
- ▶ Split

Custom keyboards

Design your own

Physical layout

- ▶ Row-staggered
- ▶ Ortholinear
- ▶ Column-staggered
- ▶ Split

Switches

- ▶ Linear
- ▶ Tactile
- ▶ Clicky

Custom keyboards

Design your own

Physical layout

- ▶ Row-staggered
- ▶ Ortholinear
- ▶ Column-staggered
- ▶ Split

Switches

- ▶ Linear
- ▶ Tactile
- ▶ Clicky

Features

- ▶ Thocky
- ▶ Clacky
- ▶ Creamy
- ▶ Marbly
- ▶ Pingy

Avalanche

Avalanche



<https://github.com/vlkv/avalanche/>

Fully Open Source: Printables (PCB, Case) etc
Full BOM & instructions available, QMK

Going further

Going further



Corne Keyboard

Going further



Corne Keyboard



Glove 80 (loan from Andreas Gulla)

Conclusion

Conclusion

Lessons Learned

Conclusion

Lessons Learned

- ▶ Great office conversation starter

Conclusion

Lessons Learned

- ▶ Great office conversation starter
- ▶ The world of keyboards goes deep

Conclusion

Lessons Learned

- ▶ Great office conversation starter
- ▶ The world of keyboards goes deep
- ▶ Hobbies can be expensive

Conclusion

Lessons Learned

- ▶ Great office conversation starter
- ▶ The world of keyboards goes deep
- ▶ Hobbies can be expensive
- ▶ Don't trust cheap Arduino USB-C sockets

Conclusion

Lessons Learned

- ▶ Great office conversation starter
- ▶ The world of keyboards goes deep
- ▶ Hobbies can be expensive
- ▶ Don't trust cheap Arduino USB-C sockets

Typing Speed

Conclusion

Lessons Learned

- ▶ Great office conversation starter
- ▶ The world of keyboards goes deep
- ▶ Hobbies can be expensive
- ▶ Don't trust cheap Arduino USB-C sockets

Typing Speed

- ▶ It's not just the keyboard

Custom Keyboards

Building your own Layouts, Firmware and Hardware

Carsten Thiel

October 23, 2026