

[swest20 s4d]

# キーボードを自作しよう

エンジニアとして成長するために

---

富士通コンピュータテクノロジーズ

石田 晴幸

# 目次

---

- 自己紹介
- 背景
- 目的
- イントロ～キーボードの世界
- 問題～考え方～施策
- キーボードを自作するための技術
- 終わる前に

# 自己紹介

項番	項目	内容
1	氏名	石田 晴幸
2	所属	株式会社富士通コンピュータテクノロジーズ
3	業務履歴	プリンタファームウェア
4		プリンタドライバ
5		携帯電話ファーム
6		ストレージのWeb GUI
7		モデル駆動開発のツール
8		二次電池の充放電試験装置
9	使用言語	C, C++, C#, Java, JavaScript, Python
10		UML, SysML
11	趣味	ジョギング・山登り・キーボード作り

※今日のプレゼンは会社とは関係ありません。個人の活動です。

# [背景] パーソナル・ファブリケーション環境の充実

---

- 3Dプリンタやレーザーカッター
  - ponoko(<https://www.ponoko.com/>)
  - FabCafe(<https://fabcafe.com/tokyo/fab/>)
- フリーで使えるCAD
  - 回路・基板設計 KiCAD(<http://kicad-pcb.org/>)
  - ケース設計 AUTODESK FUSION360  
(<https://www.autodesk.co.jp/products/fusion-360/overview>)  
※趣味で使う場合無料ライセンスを使用可能
- 個人向けPCB製造サービス
  - スイッチサイエンス(<https://www.switch-science.com/pcborder/>)
- 組み込みソフト開発環境
  - Ardiono (<https://www.arduino.cc/>)
  - Mbed (<https://www.mbed.com/>)
- 設計図の共有
  - Let's Split Guide (<https://github.com/nicinabox/lets-split-guide>)
- **自分で思い描いた組み込みシステムをそこそこのコストで実現できる！**

## [目的] 本講座の目的

---

- ソフト技術者にエレキ・ハードの世界へ踏み出す一歩の提案
  - ソフト屋だって少し手を伸ばせば  
PCBやハードが作れる
  - 他分野からソフトを俯瞰できる技術者は貴重！  
他人に言われたものを作るだけから脱却しよう
- キーボードを自作するための  
基本的な技術の習得

# [目的]なぜキーボードを作るのか

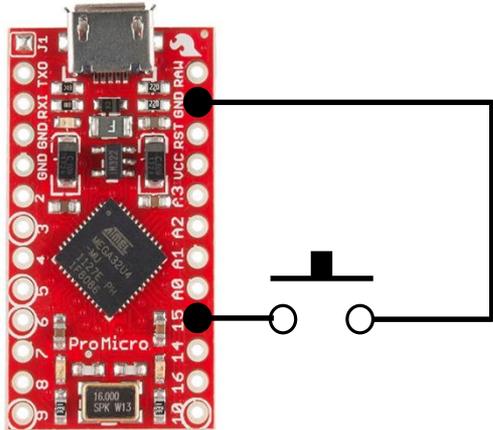
---

- 筆者：ソフト開発者として資料作成やソースコード作成の日々
  - 肩がこる
  - もっと速く正確に入力したい
  - キーボードを越える入力機器がでてこない
- HHKBは良かったけど不満もある  
明確なゴールは分からない
- 顧客は自分  
DevOpsみたいに運用しながら改善してみよう

# イントロ～ キーボードの世界

---

# イントロ.簡単なキーボードの例(1/4)



写真は <https://www.sparkfun.com/products/12640> より引用

## Arduinoのスケッチ

```
const uint8_t PIN_BUTTON = 15;
int btnStat;

void setup(){
  Keyboard.begin();
  pinMode(PIN_BUTTON, INPUT_PULLUP);
  btnStat = HIGH;
}

void loop(){
  int lastStat = btnStat;
  btnStat = digitalRead(PIN_BUTTON);
  // ボタンが押下から放された瞬間
  if ((lastStat == LOW) && (btnStat != LOW)){
    Keyboard.print("Hello");
  }
}
```

# イントロ. 簡単なキーボードの例 (2/4)

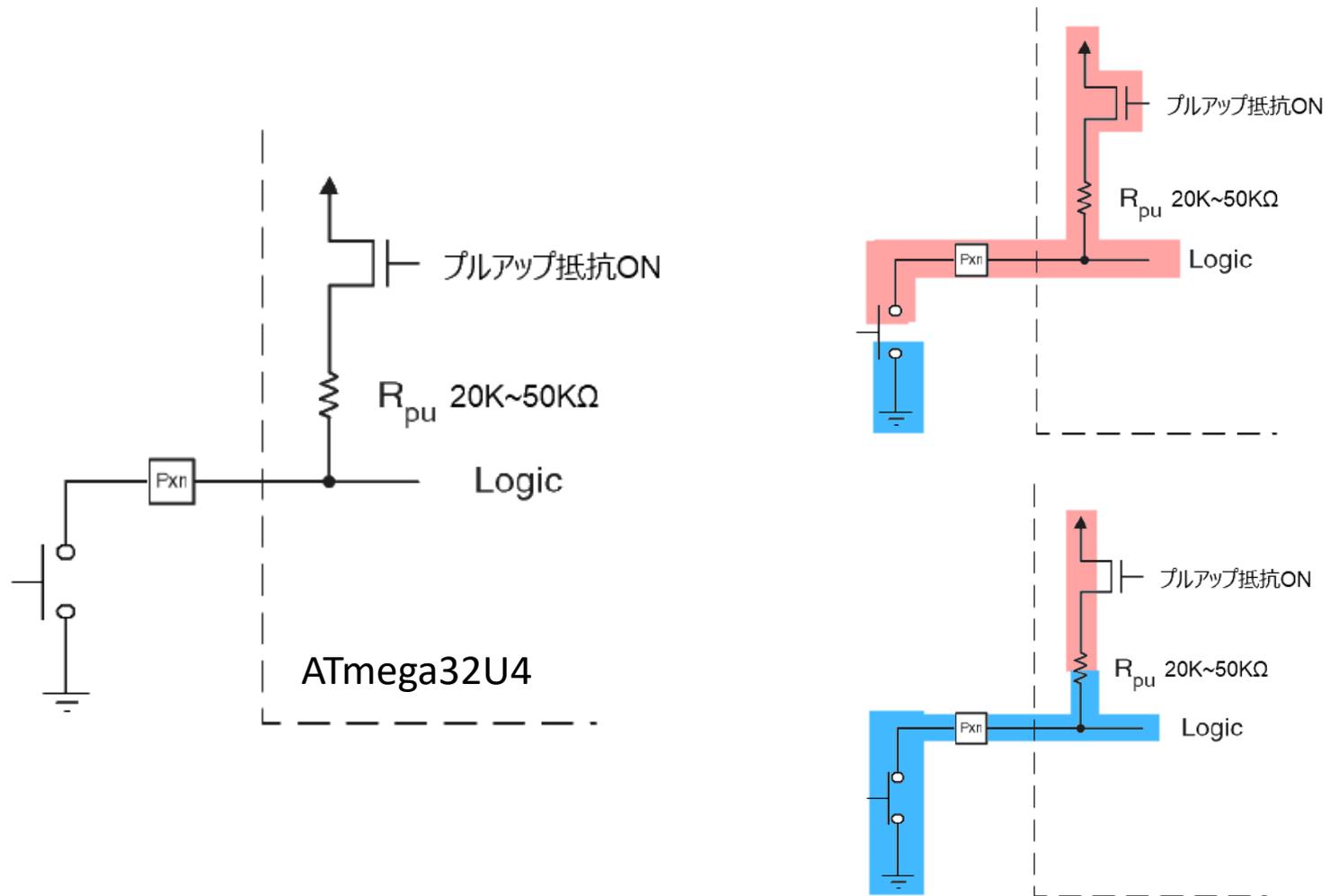
## ● AVRマイコン ATmega32U4

項番	項目	説明
1	コア	8ビットAVR
2	フラッシュ	32Kbyte
3	SRAM	2.5Kbyte
4	EEPROM	1Kbyte
5	USB対応	USB仕様改訂2.0(Universal Serial Bus Specification Rev 2.0)完全適合
6	動作電圧	2.7~5.5V
7	最大速度	8MHz/2.7V, 16MHz/4.5V

データシート: <https://avr.jp/user/DS/PDF/mega16U4.pdf>

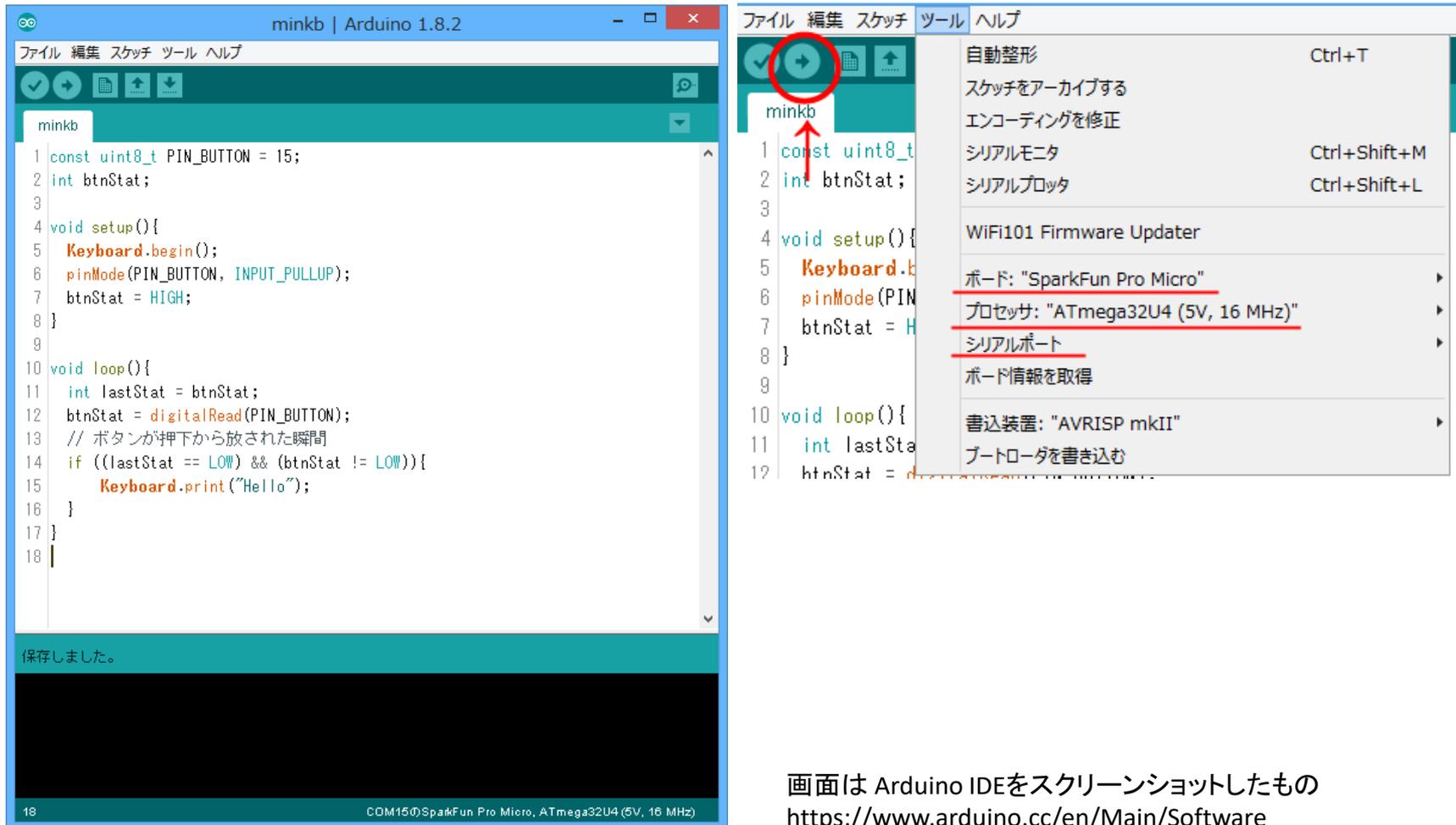
# イントロ.簡単なキーボードの例(3/4)

## マイコンによる入カピンのプルアップ



# イントロ.簡単なキーボードの例(4/4)

Arduino開発環境を用いてファームのビルドと装置への書き込み



The screenshot shows the Arduino IDE interface. The main window displays a sketch named 'minkb' with the following code:

```
1 const uint8_t PIN_BUTTON = 15;
2 int btnStat;
3
4 void setup(){
5   Keyboard.begin();
6   pinMode(PIN_BUTTON, INPUT_PULLUP);
7   btnStat = HIGH;
8 }
9
10 void loop(){
11   int lastStat = btnStat;
12   btnStat = digitalRead(PIN_BUTTON);
13   // ボタンが押下から放された瞬間
14   if ((lastStat == LOW) && (btnStat != LOW)){
15     Keyboard.print("Hello");
16   }
17 }
18
```

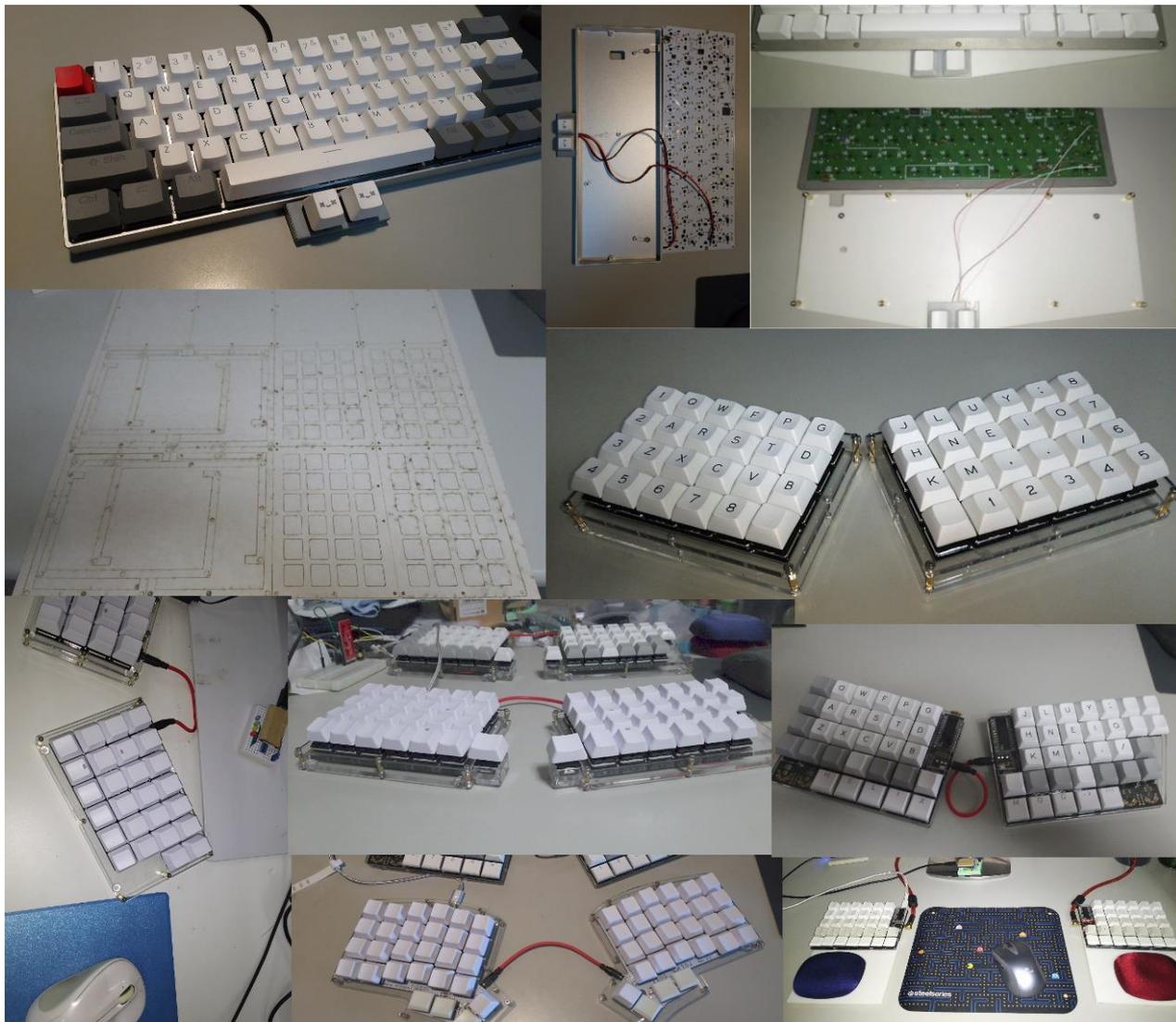
The 'Tools' menu is open, showing various options. The 'Burn Bootloader' option is highlighted with a red circle and an arrow. The menu items are:

- 自動整形 (Ctrl+T)
- スケッチをアーカイブする
- エンコーディングを修正
- シリアルモニタ (Ctrl+Shift+M)
- シリアルプロッタ (Ctrl+Shift+L)
- WiFi101 Firmware Updater
- ボード: "SparkFun Pro Micro" (expanded)
- プロセッサ: "ATmega32U4 (5V, 16 MHz)"
- シリアルポート (expanded)
- ボード情報を取得
- 書き込装置: "AVRISP mkII" (expanded)
- ブートローダを書き込む

The status bar at the bottom indicates: COM15のSparkFun Pro Micro, ATmega32U4 (5V, 16 MHz).

画面は Arduino IDEをスクリーンショットしたもの  
<https://www.arduino.cc/en/Main/Software>

# [イントロ]どんなキーボードを作ってきたのか



# キーボードの世界.フルかコンパクトか

フル



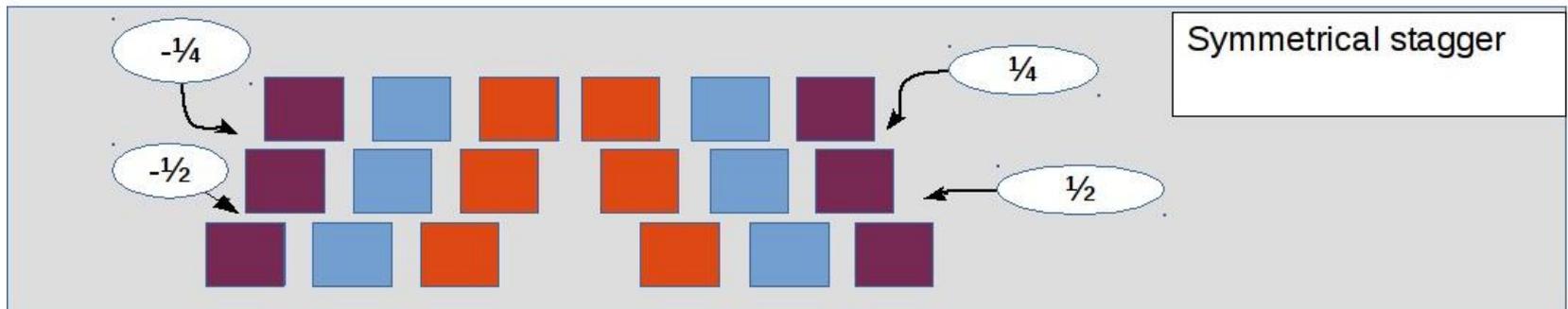
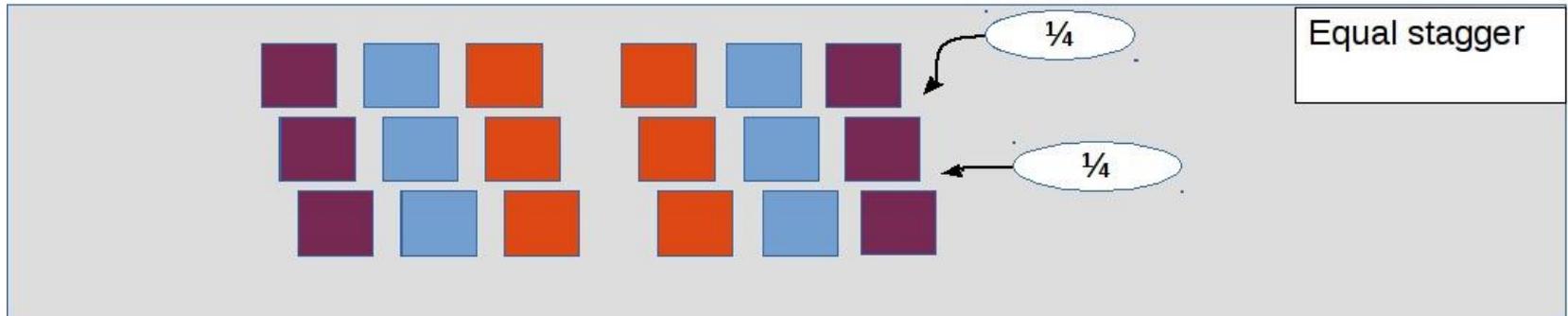
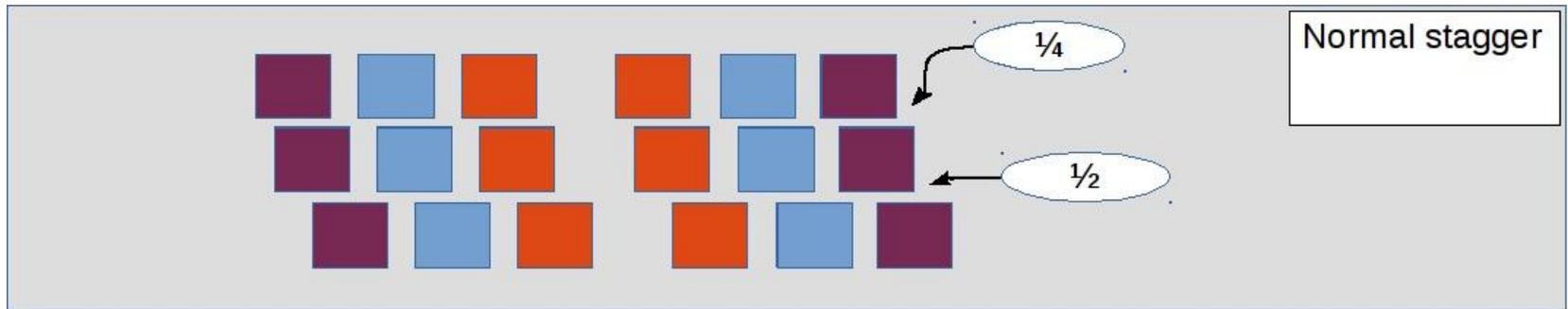
<https://upload.wikimedia.org/wikipedia/commons/5/57/ModelM.jpg> より引用

60%



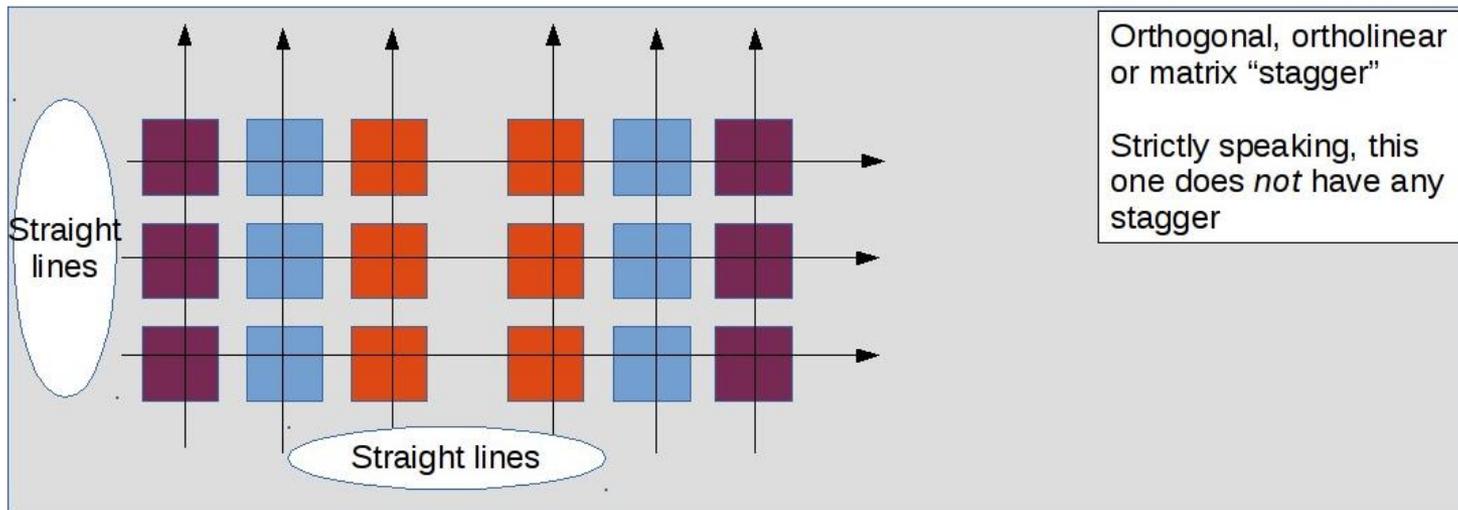
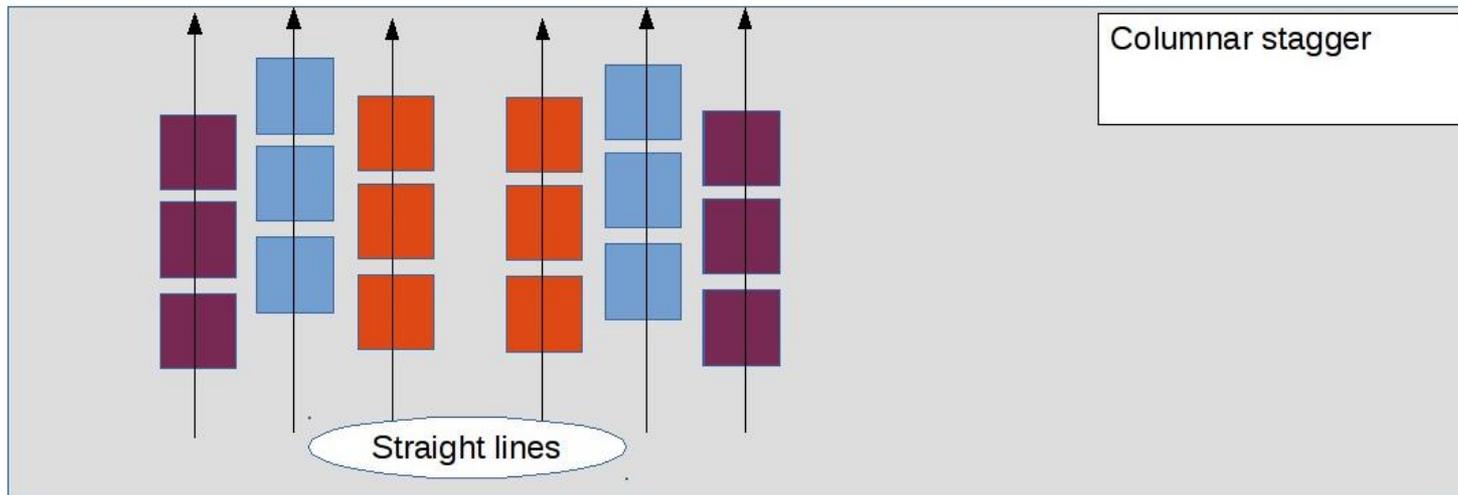
<http://www.pfu.fujitsu.com/hhkeyboard/lineup/pdkb400ws.html> より引用

# キーボードの世界. キースイッチの配置 (1/2)



<https://deskthority.net/wiki/Staggeringより引用>

# キーボードの世界. キースイッチの配置 (2/2)



<https://deskthority.net/wiki/Staggering>より引用

# キーボードの世界. 一体型か分離型か

## 一体型



## 分離型



# キーボードの世界. キー配列

## ● キー配列

項番	名称	説明
1	QWERTY	タイプライターからつづく伝統あるレイアウト。
2	Dvorak	英文の文字の出現頻度と相関性から設計。 母音を左手ホーム行に集約、 左右交互打鍵による高速入力を狙う。 1943年に発表された。
3	Colemak	QWERTYから17キーを変更。 頻度の高い文字をホーム行に集約。 AZXCVなど良く使う編集用キーは変更なし。 2006年に発表された。
4	Minimak	QWERTYから段階的に移行できる。 最小の変更で最大の利益を狙う。

# キーボードの世界.キー配列.QWERTY

~ 、	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	( 9	) 0	- _	+ =	← Backspace
Tab ↔	Q	W	E	R	T	Y	U	I	O	P	{ [	} ]	 \ _
Caps Lock ↑	A	S	D	F	G	H	J	K	L	: ;	" '	Enter ↵	
Shift ↑	Z	X	C	V	B	N	M	< ,	> .	? /	Shift ↑		
Ctrl	Win Key	Alt							Alt	Win Key	Menu	Ctrl	

<https://en.wikipedia.org/wiki/QWERTY>より引用

## 伝統をいまに引き継ぐ

わざと打ちにくくした、というのはデマらしい

# キーボードの世界.キー配列.DVORAK

~ 、	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	( 9	) 0	{ [	} ]	← Backspace
Tab ↔	" '	< ,	> .	P	Y	F	G	C	R	L	? /	+ =	 \ _
Caps Lock ↑	A	O	E	U	I	D	H	T	N	S	- _	Enter ↵	
Shift ↑	:	Q	J	K	X	B	M	W	V	Z	Shift ↑		
Ctrl	Win Key	Alt							Alt Gr	Win Key	Menu	Ctrl	

## 左右の手で交互打鍵、高速入力を狙う

Linuxの人はls -l がきつい(小指多用)  
日本語の「か行」ローマ字入力で交互打鍵が崩れる

# キーボードの世界.キー配列.COLEMAK

~ ,	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	( 9	) 0	- _	+ =	← Backspace
Tab ↔	Q	W	F	P	G	J	L	U	Y	:	{	}	 \
← Backspace	A	R	S	T	D	H	N	E	I	O	"	'	↵ Enter
Shift ⬆	Z	X	C	V	B	K	M	< ,	> .	? /	Shift ⬆		
Ctrl	Win Key	Alt								Alt Gr	Win Key	Menu	Ctrl

<https://en.wikipedia.org/wiki/Colemak>より引用

## 最強なのは？

Ctrl-A, Ctrl-Z, Ctrl-X, Ctrl-C, Ctrl-VがQWERTYと同じ親切設計  
ホーム行の高ヒット率で入力を効率化する

# キーボードの世界.キー配列.minimak

## ●minimak



QWERTYから4キー変更



QWERTYから8キー変更



QWERTYから12キー変更

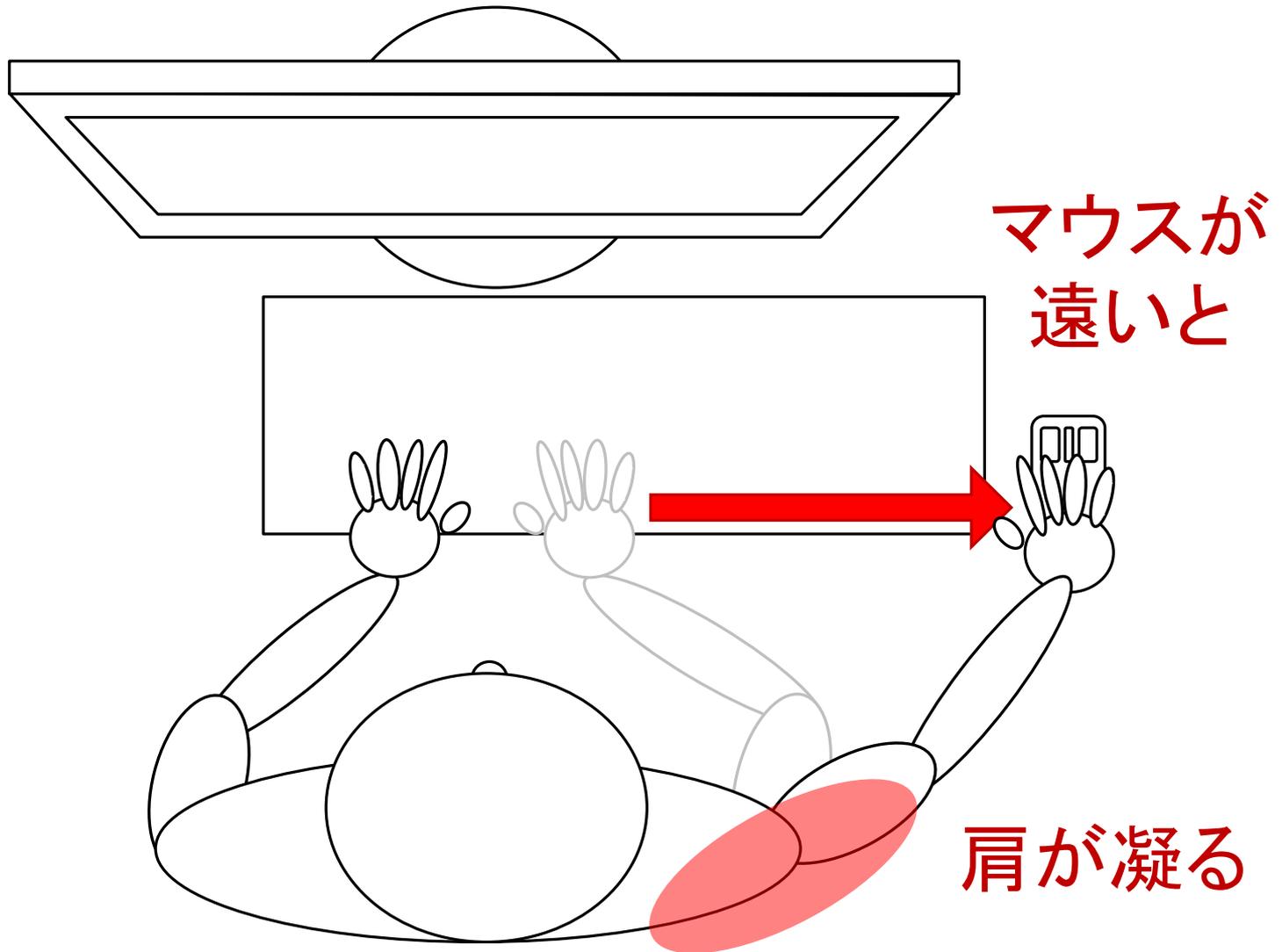
画像は<http://www.minimak.org>より引用

## QWERTYから段階的に移行できる

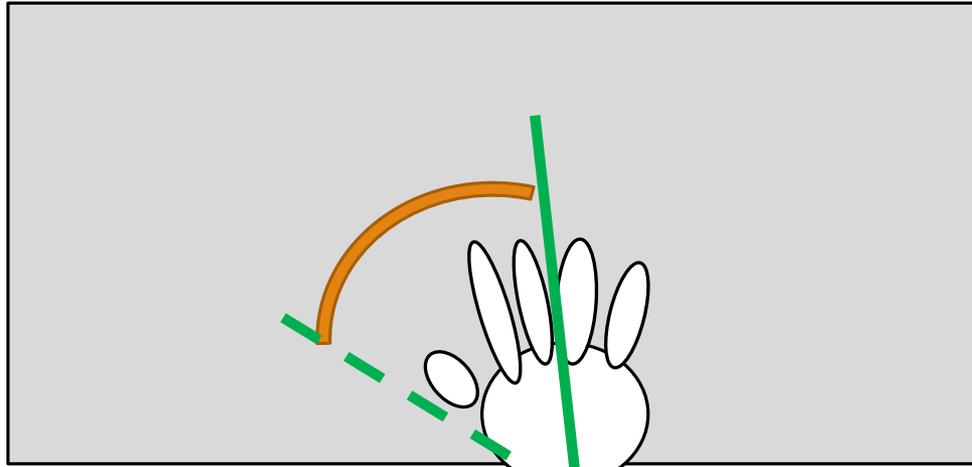
# 問題 考え方 施策

---

# 問題. 遠いマウス



# 問題. 手首の角度



手首に角度を  
つけると

肩・腕が  
疲れる

# 問題.小指の酷使



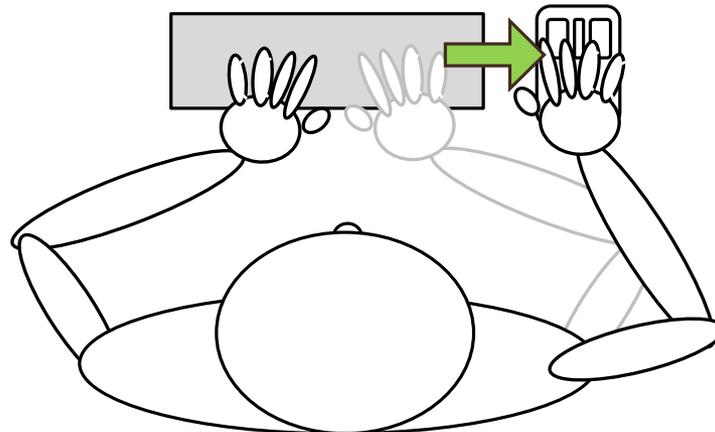
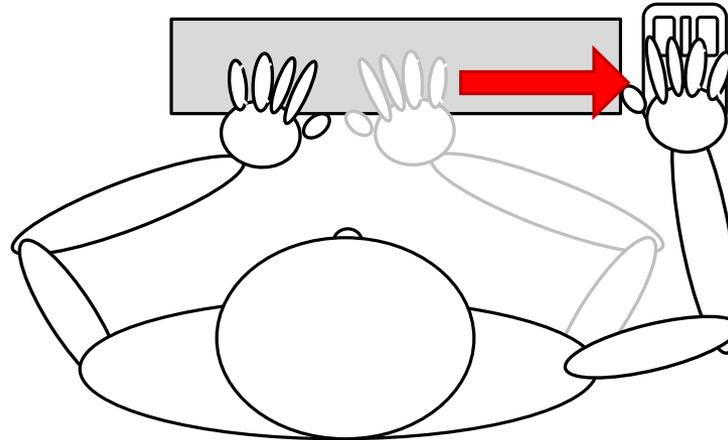
小指を酷使すると  
前腕が疲れる

# 考え方. マウスを近くに

テンキーレスでマウスを  
ホームポジションの近く  
に置く

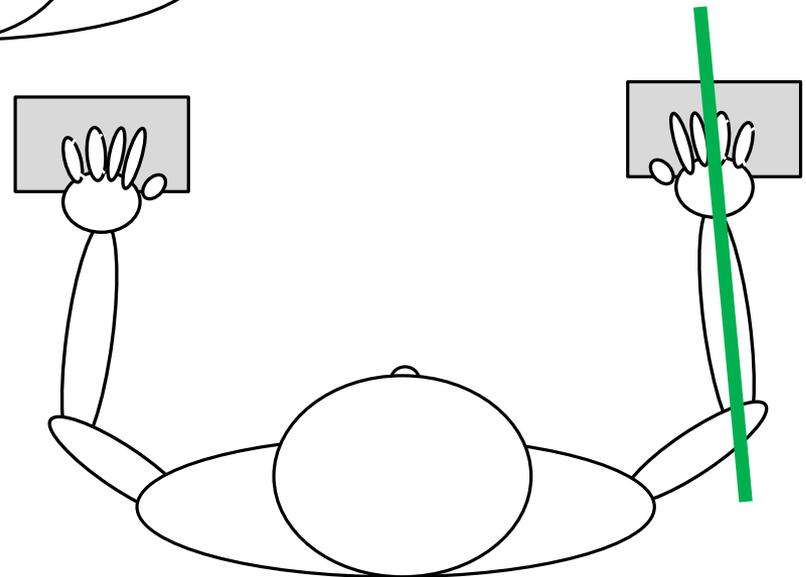
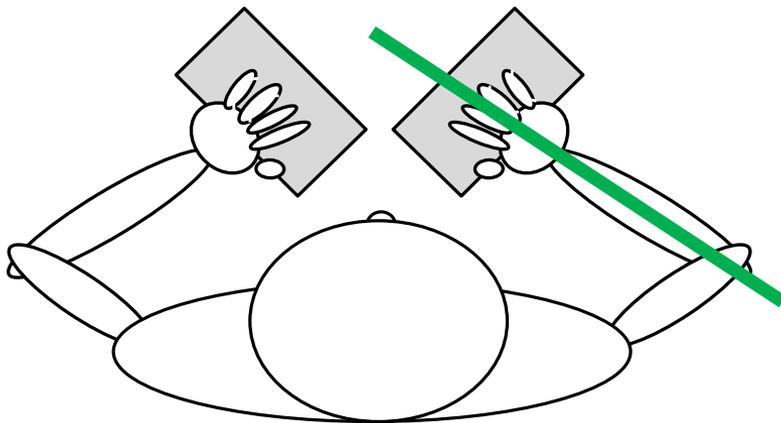
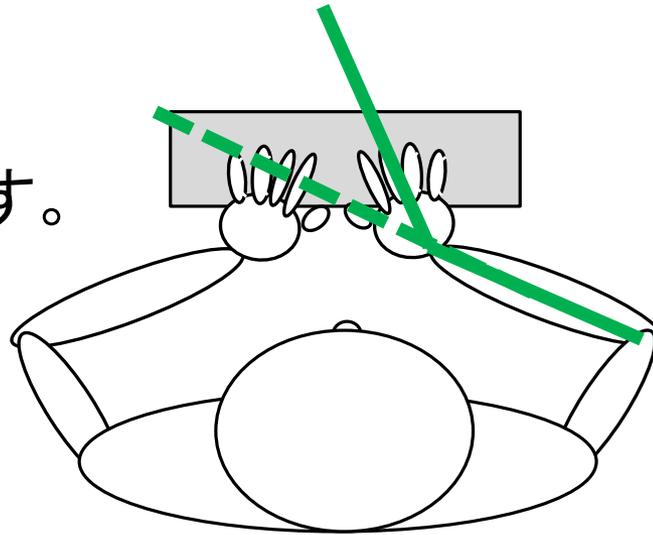
カーソルキーなどはレイ  
ヤキーとの同時押しで、  
ホームポジションのまま  
使えるようにする。

※レイヤキーについては後  
述します。



# 考え方.手首の角度の改善

分割キーボードで  
手首の角度をなくす。

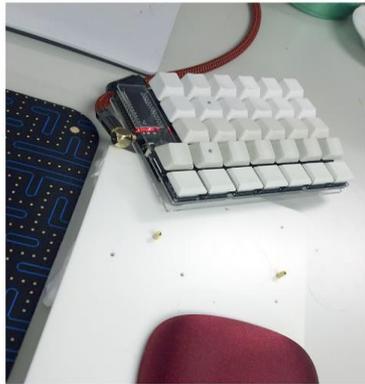


# 考え方. モディファイアキーは親指で



親指に力を入れても、  
残りの指は自由に動ける

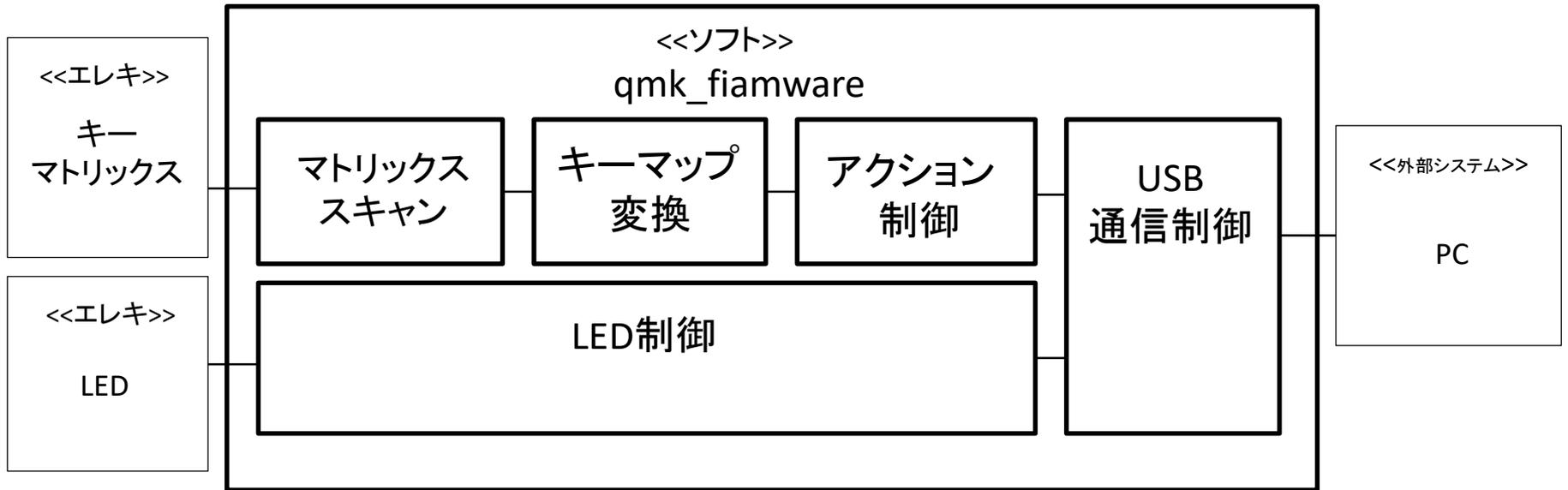
# 施策.現在のキーボード



# キーボードを 自作するための 技術

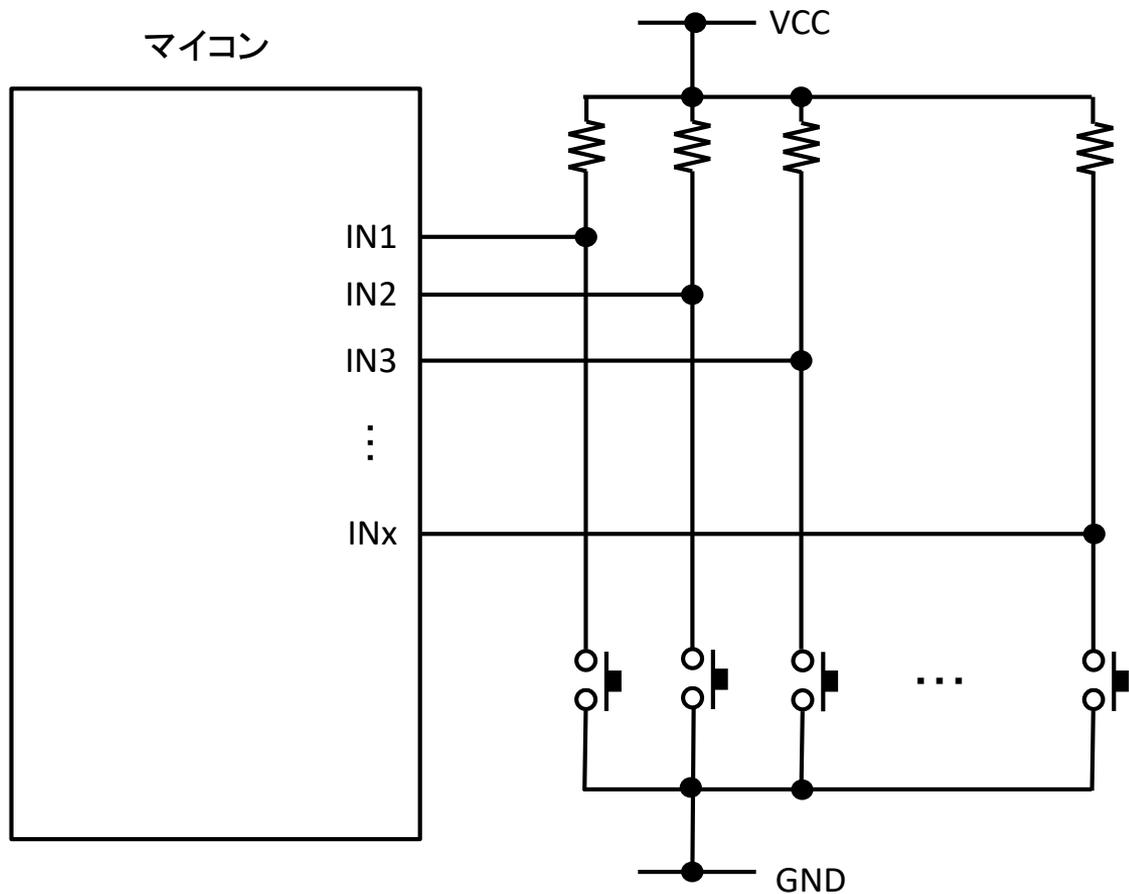
---

# 技術.全体構造



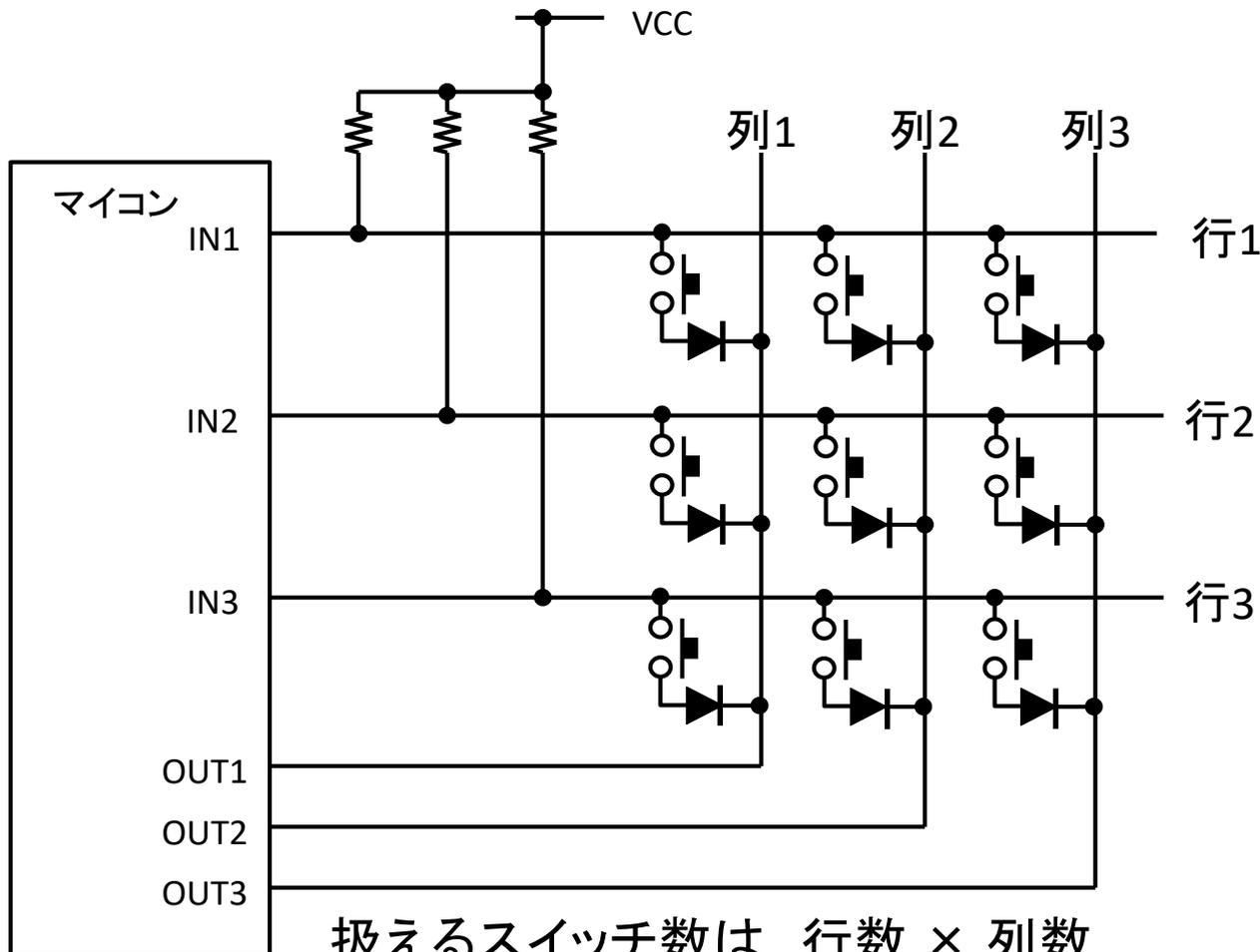
# 技術.エレキ.スイッチマトリックス(1/2)

簡単キーボードを  
そのまま拡張すると...



スイッチの数だけマイコンのピンが必要？  
100個のスイッチなら100本の入力ピン！？

# 技術.エレキ.スイッチマトリックス(2/2)

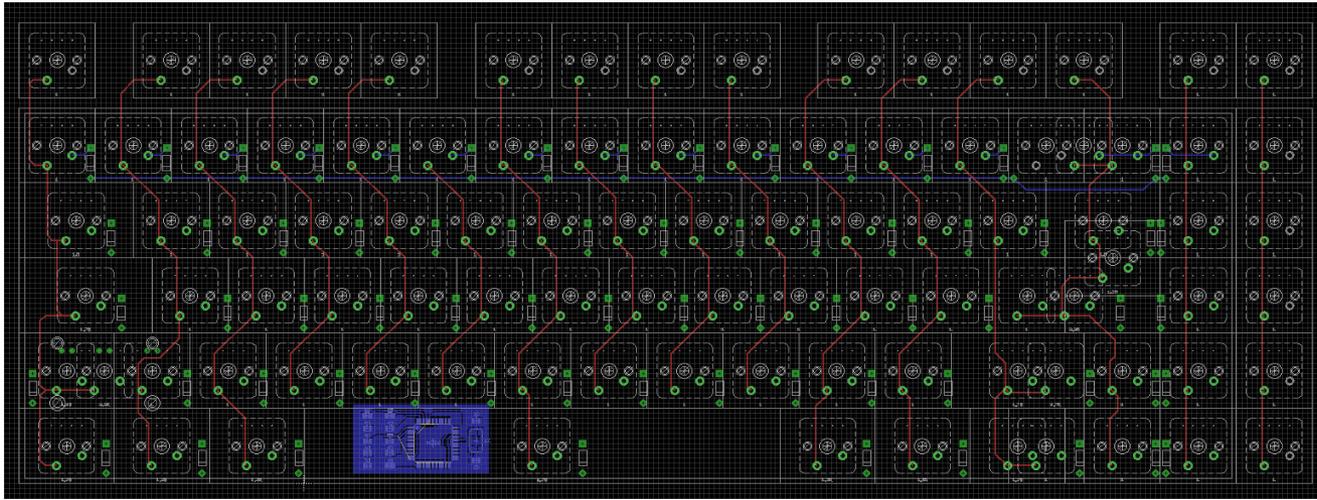


扱えるスイッチ数は 行数 × 列数

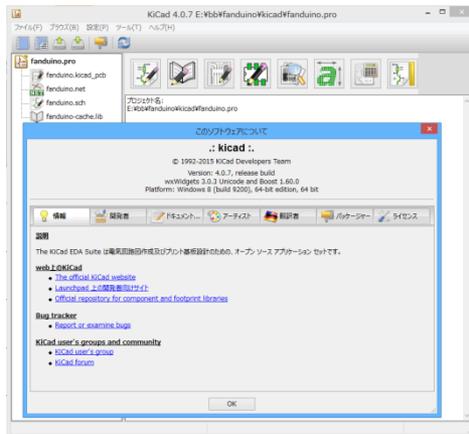
必要なマイコンのピン数は 行数 + 列数

例:100個のスイッチは (10+10)=20本のピンでOK!

# 技術.エレキ.PCB



<https://deskthority.net/keyboards-f2/inexpensive-pcb-custom-boards-t12065.html> より引用



KiCad (<http://kicad-pcb.org/>)

## google で pcb 個人 製作 で検索

A's Lab II : 個人レベルの自作プリント基板の製作

[haido.blog.jp/archives/1005436376.html](http://haido.blog.jp/archives/1005436376.html)

2014/07/04 - 個人レベルの自作プリント基板の製作。やっと期末テストが終わり落ち着いているHaidoです。手ごたえは中々いい感じでテストの返却が楽しみです。久々の更新と言うことで色々書いたら文字だらけの記事になってしまったので要らない部分は...

1万円プリント基板試作・量産製造のunicraft (法人・個人向け) -株式会社...

<https://unicraft-jp.com/pcb/>

プリント基板の設計・製造・通信販売の株式会社ユニクラフト！イニシャル費用無料で総額1万円以下から製作可能！法人のお客様から個人のお客様まで大歓迎！

個人でも使えるプリント基板業者 - NAVER まとめ

<https://matome.naver.jp/odai/2138026954372116601>

2016/07/27 - 最近ではプリント基板を安価に作ってもらえる業者が増え、個人でも利用できる時代になりました。個人からの注文も受け付けてくれる業者をまとめてみました。全てネット... 設計から製作までプリント基板の事なら「きばん本舗」にお任せ下さい。

プリント基板をつくってみる〜Fusion PCBレビュー〜 | MUDAなこと...

<https://make-muda.net/> | 電子工作 | その他

2017/07/16 - 自分は本格的に電子工作を始めて3年目ぐらいですが、今回初めて、自分の作品に自作プリント基板を導入しました... というこで、随分前置きが長くなりましたが、個人で使えるプリント基板制作サービス「Fusion PCB」の利用レビューです。

# 技術. ハード. スイッチ

- 種類

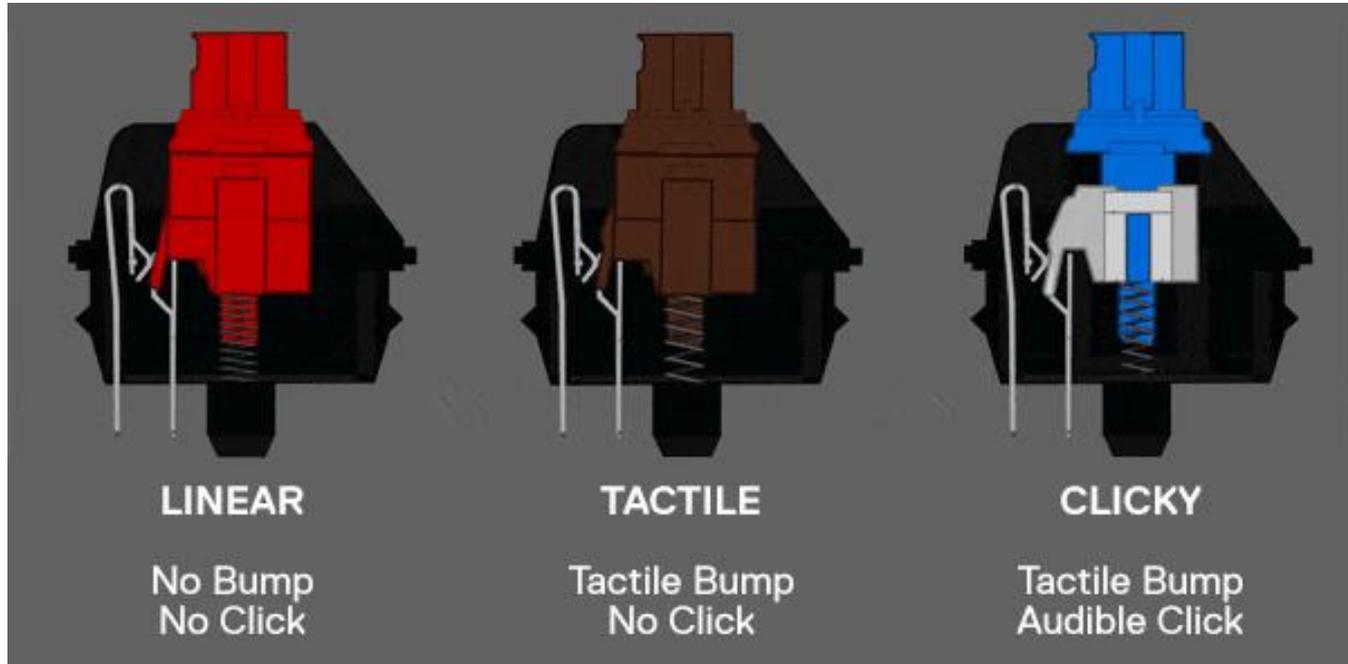
- メンブレン
- メカニカル
- パンタグラフ
- 静電容量
- etc..

- メカニカルスイッチの構造

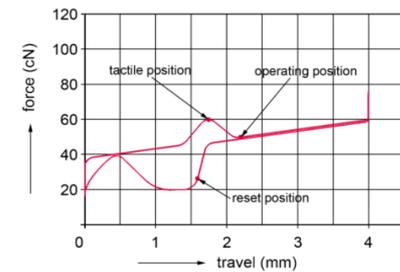
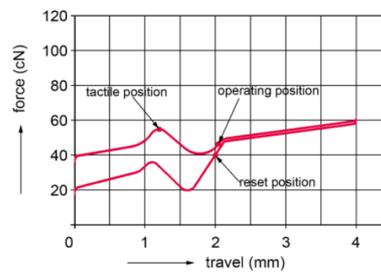
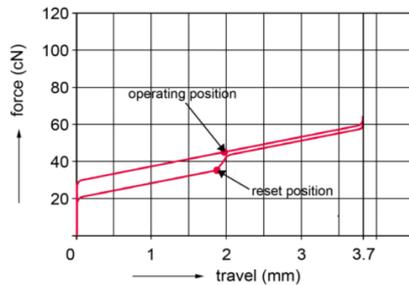


<https://sketchfab.com/models/4f4cfbf9f50949b193ab99e43f4ca23d>より引用

# 技術. ハード. スイッチ. 使用感

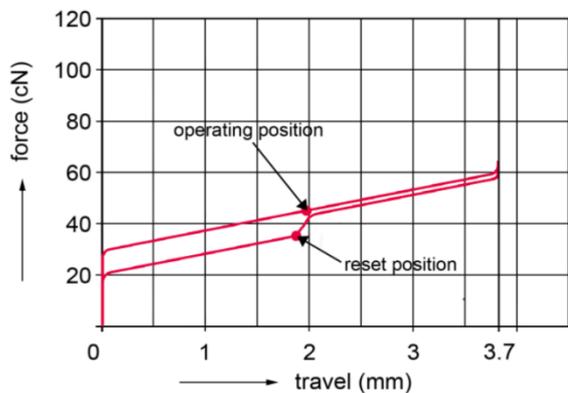


<https://jp.steelseries.com/blog/gaming-keyboard-mechanical-switches-44>より引用

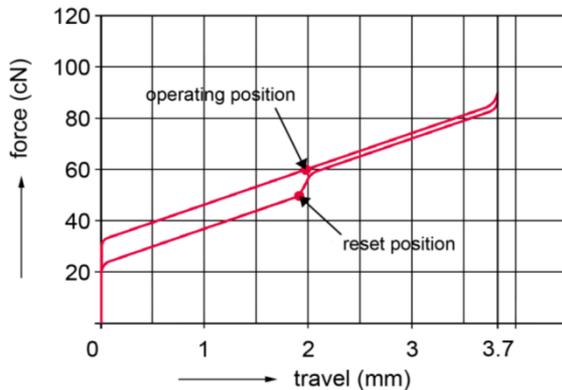


<http://cherryamericas.com/product/mx-series-2/>より引用

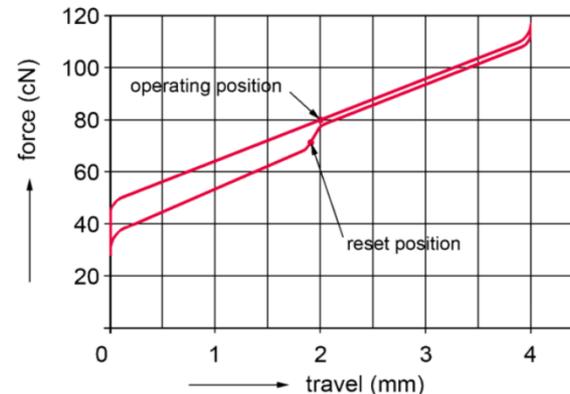
# 技術. ハードスイッチ. 作動力



45cN

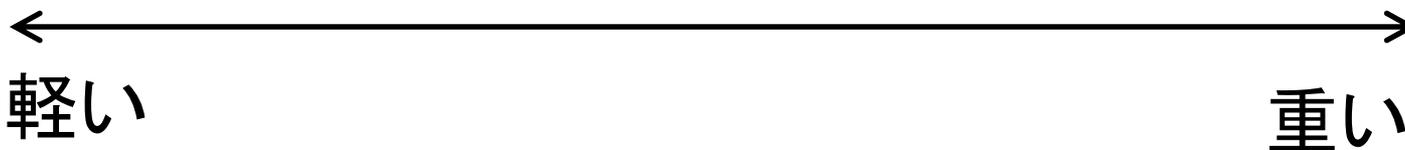


60cN



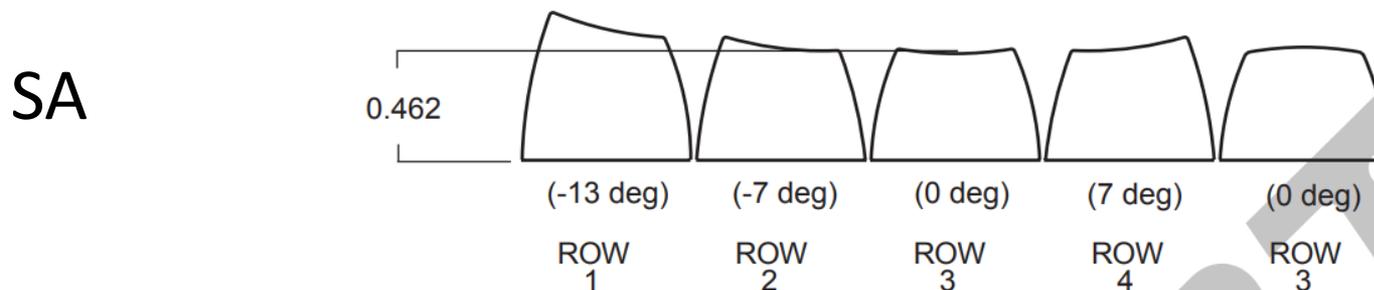
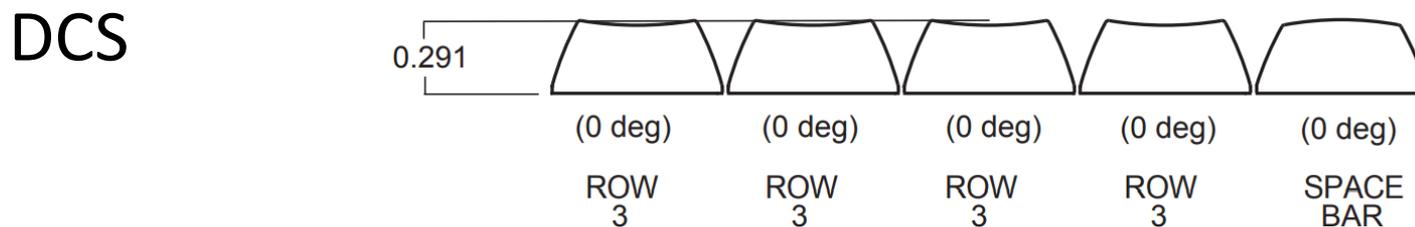
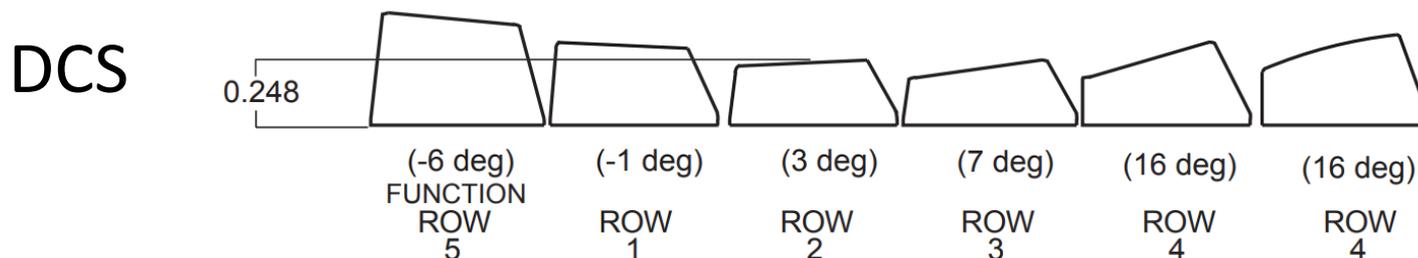
80cN

<http://cherryamericas.com/product/mx-series-2/>より引用



# 技術. ハード. キーキャップ (1/4)

## ● プロファイル



# 技術. ハード. キーキャップ (2/4)

## ●材質

項番	材質	耐熱温度	価格	特徴
1	ABS	100°C	安価	色が多い
2	PBT	140°C	高価	サラサラ感 昇華プリント可

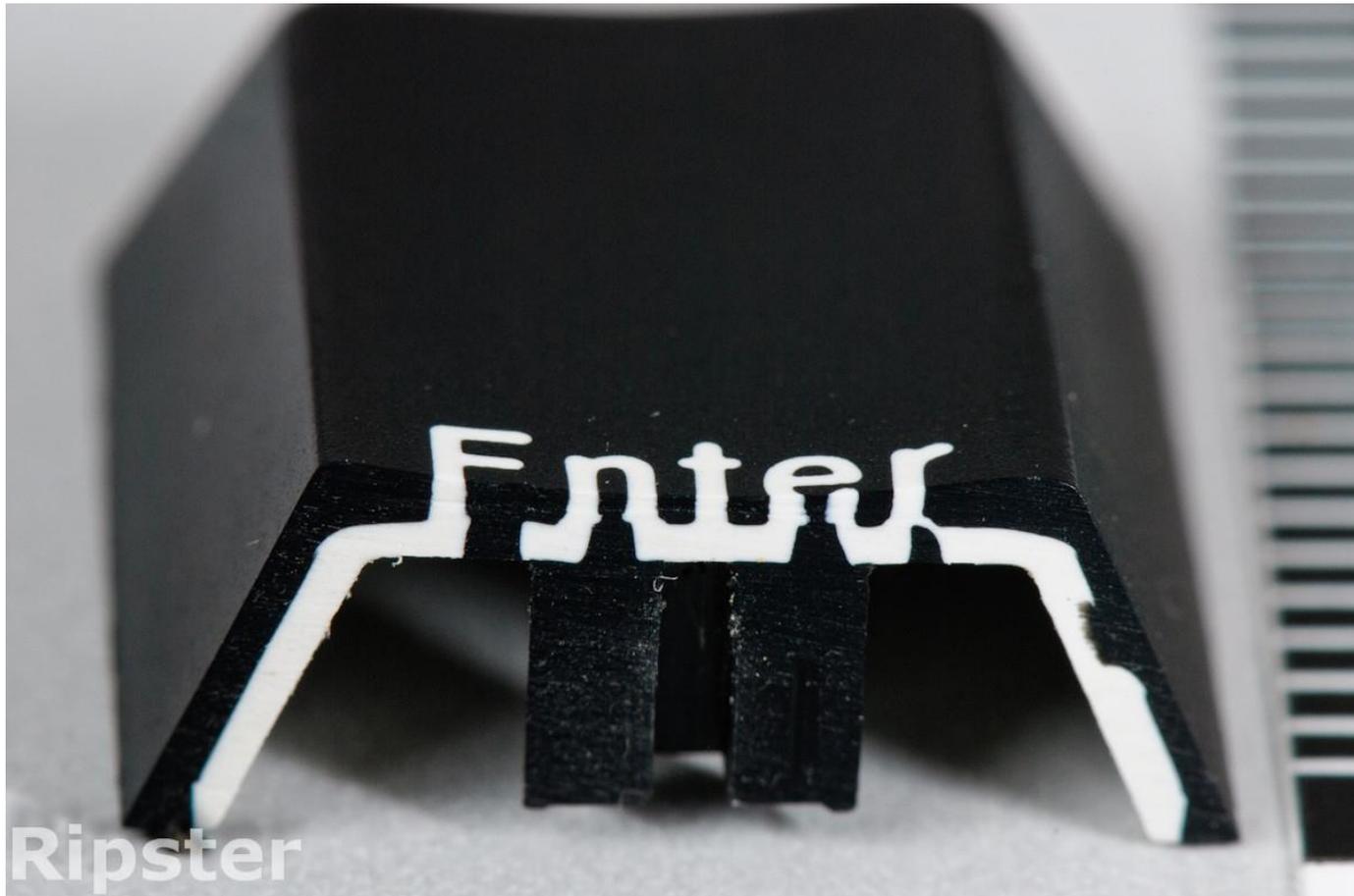
## ●加工法

項番	加工法	価格	特徴
1	プリント	安価	擦れて文字が消える
2	ダブルショット	高価	文字が消えない バックライトで映える

# 技術. ハード. キーキャップ (3/4)

---

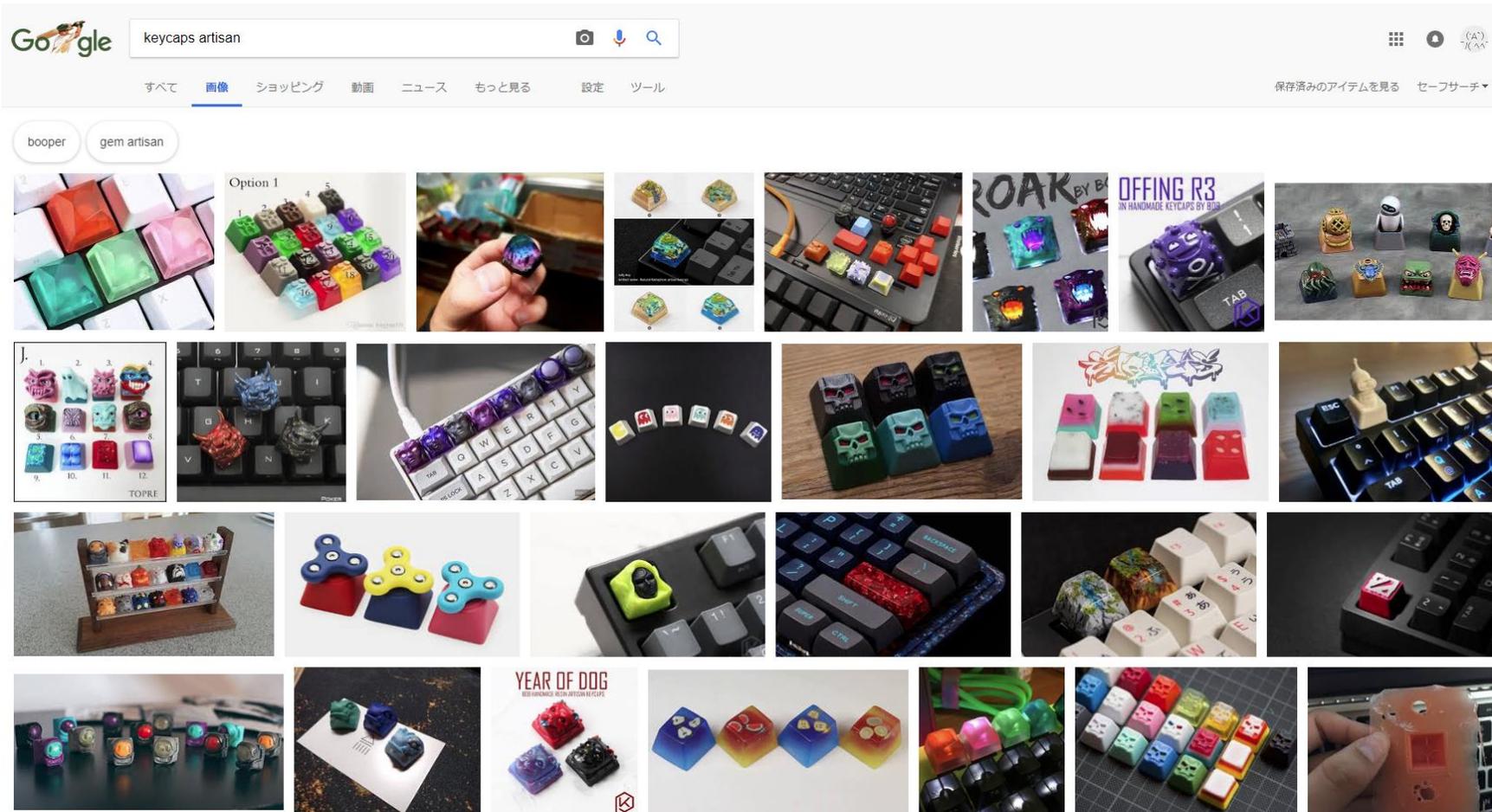
- 加工法. ダブルショット



<https://imgur.com/a/OhRd4#NV570xa> より引用

# 技術. ハード. キーキャップ (4/4)

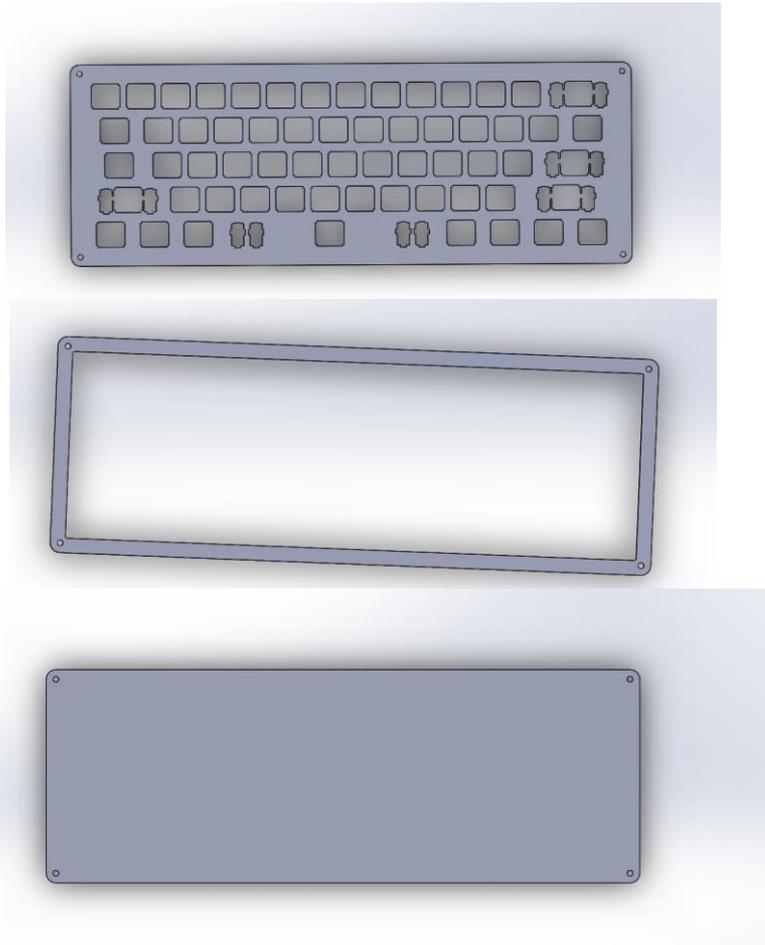
## ●職人



google で keycaps artistan で検索した結果

# 技術. ハード. ケース

## ● アクリル板



<https://imgur.com/a/fOSwe> より引用

## ● 3Dプリント



<https://www.thingiverse.com/thing:983332> より引用

# 技術.ソフト.qmk\_firmware



The goal of the QMK software project is to develop a completely customizable, powerful, and enjoyable firmware experience for any project - keyboard or otherwise - and to provide helpful, encouraging, and kind support and feedback for people with any software development experience.

[View on !\[\]\(0b9e2e386fe163003656c0c1fc02970c\_img.jpg\) GitHub](#)

download [zip](#) download [tar](#)

tag [planck-6.0](#) build [passing](#) chat [199 online](#) docs [ready](#) contributors [377](#)

 Fork [5239](#)

<https://qmk.fm/>より引用

# 技術. ソフト.qmk\_firmware. マトリックススキャン

- 設計パラメーター(キーボードごとに違う)  
ソース qmk\_firmware¥keyboards¥<keyboard名>¥config.h
  - 行を選択して列をセンス / 列を選択して行をセンス  
#define DIODE\_DIRECTION COL2ROW
  - 行・列とマイコンのGPIOピンとの対応  
#define MATRIX\_ROW\_PINS { D0, D1, D2, D3, D5 }  
#define MATRIX\_COL\_PINS { F0, F1, E6, C7, C6, B6, D4, B1, B7, B5, B4, D7, D6, B3 }
  - デバウンスの有無・時間  
※スイッチがONになる瞬間に電気信号がHi/Loを繰り返す  
信号が変化してから安定するまでセンスしない(だいたい5msec)  
#define DEBOUNCING\_DELAY 5
- ソース: qmk\_firmware¥quantum¥matrix.c
  - 初期化 matrix\_init()
  - スキャン実行 matrix\_scan()

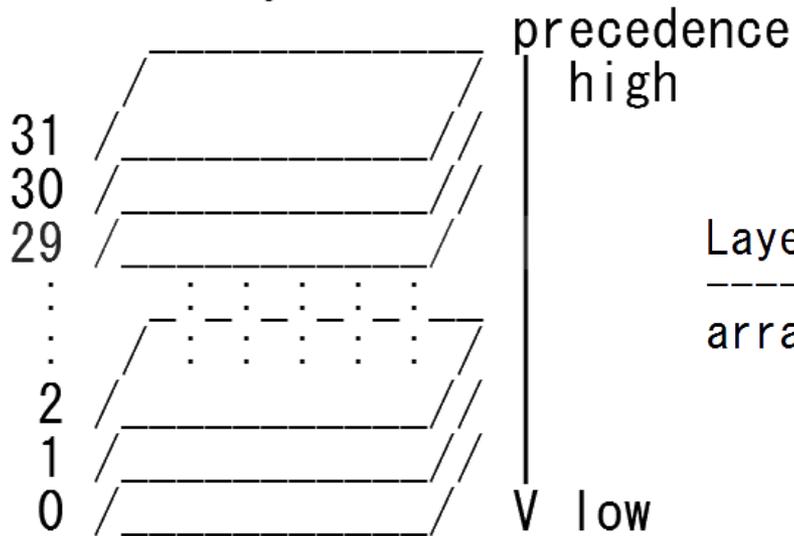
# 技術. ソフト.qmk\_firmware. キーマップ

- ソース `qmk_firmware/keyboards/<キーボード名>/keymaps/<キーマップ名>/keymap.c`
  - ※数えたら987個のキーマップファイルがありました。  
どんどん増えていきます。
  - キーマップ定義
    - `const uint16_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS]`
- キーマップの構造
  - レイヤーごとに繰り返す (後述します)
    - マトリックスの行ごとに繰り返す
      - マトリックスの列ごと繰り返す
        - キー1つの定義
- キー1つの定義
  - 通常のキー
  - モディファイア操作 + 通常キー
  - アクション (後述します)

# 技術. ソフト.qmk\_firmware.キーマップ.レイヤー

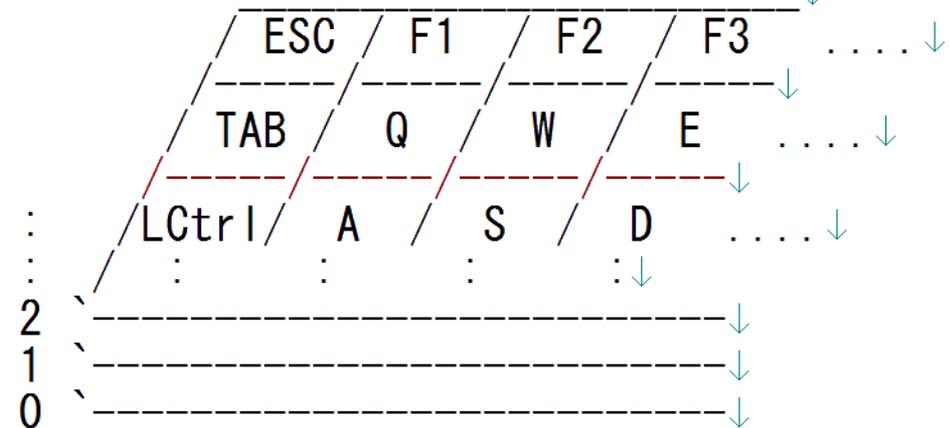
Keymap: 32 Layers

-----  
stack of layers



Layer: action code matrix↓

-----↓  
array\_of\_action\_code[row][column]↓



[https://github.com/qmk/qmk\\_firmware/blob/master/docs/keymap.md](https://github.com/qmk/qmk_firmware/blob/master/docs/keymap.md) より引用

# 技術. ソフト.qmk\_firmware. アクション

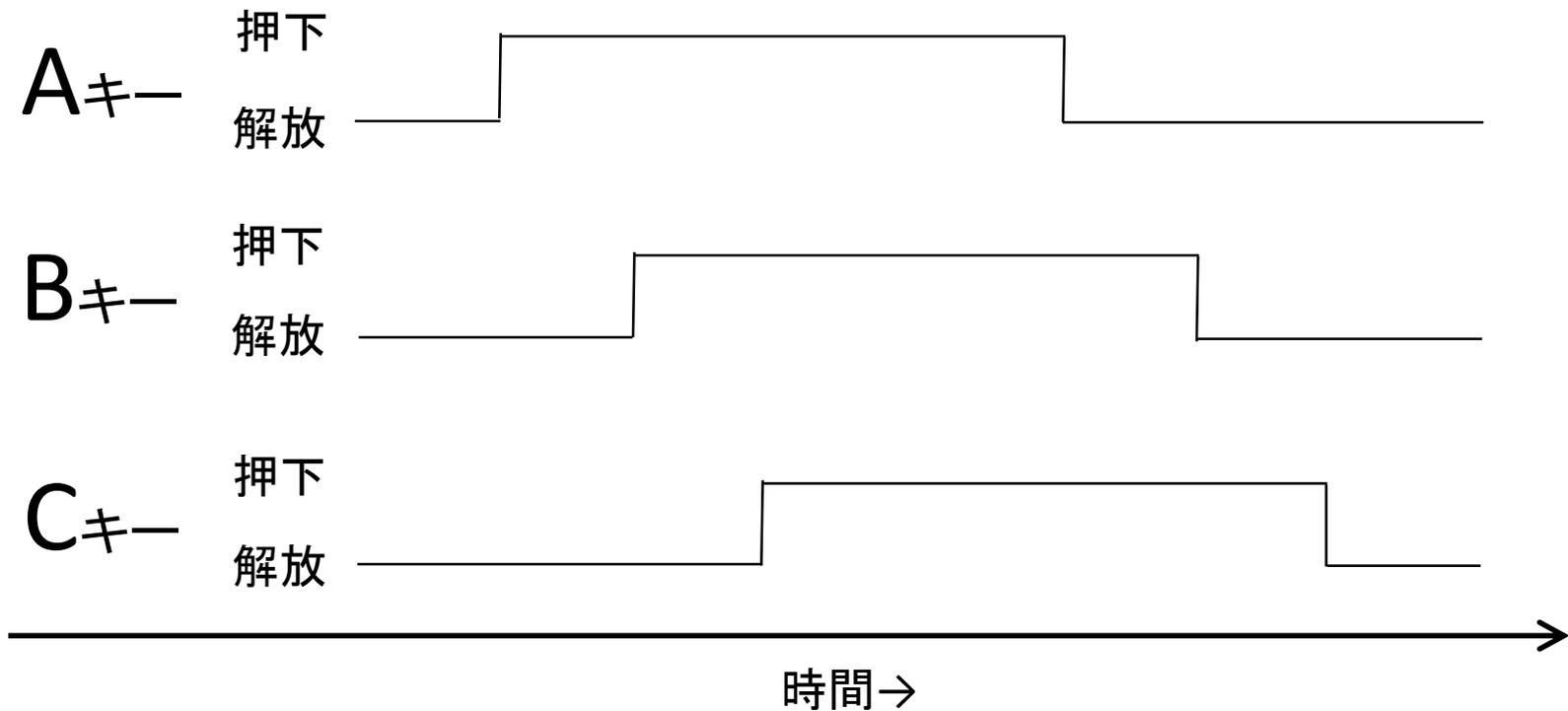
- キーを押されたとき、キーボード内の状態を変えり特殊な動作をする
- アクションの種類
  - レイヤー操作
    - レイヤー有効・無効にする
    - あるキーが押されている間だけレイヤーを有効にする
    - レイヤーの有効・無効をトグルする
    - あるキーと同時に他のキーが押されたらレイヤーキーとして扱う、あるキーが単体で押下・開放されたら通常キーとして扱う
    - 次のキーだけレイヤーを有効にする
  - バックライト操作
    - バックライト点灯・消灯する
    - バックライト点灯・
    - バックライト輝度の増減
    - バックライト点灯・消灯のトグル
  - その他

# 技術. ソフト. USB通信

---

- USB HID Keyboardのディスクリプタ
  - USBプロトコルの詳細に入り込むと今日は帰てなくなるので要点だけ
  - USBキーボードは、キーが押された・放されたとき、USBホストへレポートを送る。
  - レポートの形式はレポート・ディスクリプタとして定義する。  
(レポートというデータの構造を表現するデータ)
  - ただし、PC起動時(BIOS)はBOOT キーボードプロトコルで通信する。  
このときのレポート形式は決め打ち。  
(ここではBOOTキーボードレポート形式と呼ぶ)
- レポート・ディスクリプタを決める要因
  - Nキーロールオーバー  
(いくつのキーまで同時押しを扱えるのか)
  - BOOT Reportは6キーロールオーバー

# 技術.Nキーロールオーバー



ABCと入力するときに、前のキーを放す前に次のキーを押す  
速く入力するとき、ひとはこのようにキーを操作する  
同時押しを扱える最大キー数がN

# 技術. ソフト. USB通信. レポート形式

## 6キーロールオーバー/ Bootキーボードレポート



押されているキーのコード群(最大6)  
空きは0x00

モディファイア

- bit0: 左Ctrl
- bit1: 左Shift
- bit2: 左Alt
- bit3: 左Gui (Windowsキー等)
- bit4: 右Ctrl
- bit5: 右Shift
- bit6: 右Alt
- bit7: 右Gui (Windowsキー等)

終わるまえに

---

# etc. 自作キーボード界隈情報

---

- Self Made Keyboard in Japan(日本語)

<https://discordapp.com/channels/376937950409392130/376937950409392132>

- Twitter #自作キーボード

<https://twitter.com/hashtag/自作キーボード>

- r/MechanicalKeyboards

<https://www.reddit.com/r/MechanicalKeyboards/>

- geekhack

<https://geekhack.org/>

- deskthority

<https://deskthority.net/>

# まとめ

---

- 自作キーボードのとりかかりは簡単  
Pro Microボードとスイッチ。Arduino IDEは無料
- 情報はインターネットに散在している  
ハードのCADデータも、PCBのCADデータも、ファームのソースも  
github等にある
- コミュニティも活発
  - 海外の掲示板で世界中のオタクと交流  
(筆者も英語で書き込んだら、ドイツ人に褒めて貰えた)
  - 日本語の掲示板もできた  
(ベトナムからの留学生が活発に発言してた)

日々使うものを自分で改善  
新しいものへのチャレンジの日々