

[swest20 s4d 予稿]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

1. はじめに

3D プリンター・レーザーカッターの普及、ガーバーデータ入稿によるプリント基板製造サービスの低価格化等、少量でも安価に高品質なものづくりの環境が整ってきました。

一方、会社勤めのエンジニアにはこのような社会の変化を実感できていない人が多いのではないのでしょうか？

本セッションでは、日頃よく使う入力装置であるキーボードの自作を通じ、メカ・エレキ・ソフトを見通せるエンジニアに成長する一歩となることを狙います。

組込みソフトの開発者は一日のうちかなりの時間をキーボード入力に費やしていると思います。つまりキーボードの利用者ということです。

いまお使いになられているキーボードにとくに不満を感じていない人も多いかと思います。しかし、キーボードを変えることで、肩こりが低減したり、もっと効率的に文字列を入力出来たりするかもしれません。

自分自身を顧客とし、目標に向かってメカ・エレキ・ソフトの分野に跨って柔軟に思考し短いスパンで PDCA を回す、製作したキーボードを使うなかで挙がった問題点を次の開発につなげる DevOps の実践、それが自作キーボードの楽しさです。こういった行動を身につけることで、これまでと違った観点から実務も見ることができるようになることでしょう。

2. キーボードというシステム

キーボードはパソコンの入力装置のひとつですが、キーボード自体もマイコンを内蔵したシステムとなっています（図 2-1）。最近のキーボードには ARM Cortex M3 を搭載しているものまであります。

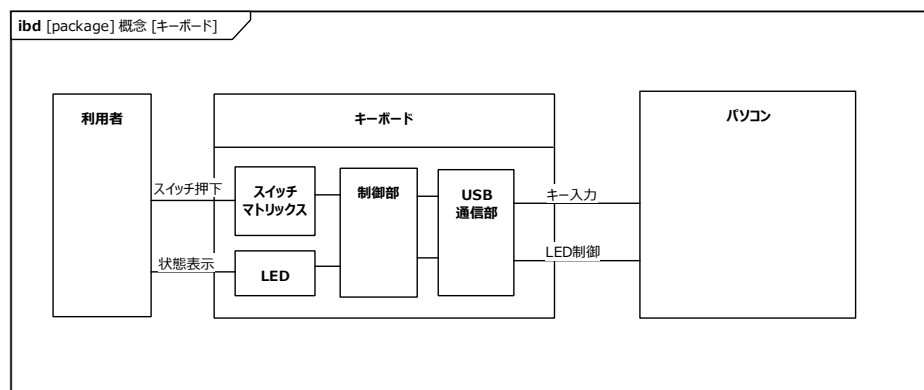


図 2-1 キーボードというシステム

表 2-1 にキーボードの構成要素を示します。

メカ・エレキ・ソフト 3 分野の要素が入っています。しかし、それぞれの要素を取り扱うのにそれほど高度な技術は要りません。ソフトウェア技術者が他の分野へ手を伸ばすのに良い題材と言えます。

表 2-1 キーボードの構成要素

項番	分野	要素
1	メカ	ケース
2		USB コネクタ
3		(スイッチのホットスワップ用ソケット)
4		(ユニット間通信用ケーブルのソケット)
5	エレキ	スイッチマトリックス
6		マイコン周りの配線
7		(ユニット間通信用配線)
8		(LED 駆動用 FET)
9	ソフト	スイッチマトリックスのスキャン
10		キーコード生成
11		USB キーボード・レポートの送信
12		(ユニット間通信処理)
13		(LED 制御)

※0で括った要素は一部のキーボードで必要となるもので必須ではありません。

3. キーボード開発事情

2017 年秋、日本において自作キーボード製作者がいきなり増えたように思います。その一因は Let's Split キーボードでしょう(図 3-1)。Let's Split は 6×4 の格子配列ユニットが 2 つからなるキーボードで、その突飛な見た目と比較的安価なキットが販売されたことから注目されました。



図 3-1 Let's Split キーボード

<https://github.com/nicinabox/lets-split-guide> より引用

Let's Split に限らずオープンハードウェアとして開発されているキーボードは多数あり github など回路図や CAD データが公開されています。

Let's Split で使われているマイコン基盤は sparkfun Pro Micro (<https://www.sparkfun.com/products/12640>) やその互換ボードで ATmega32U4 チップが搭載されています。ATmega32U4 には USB が内蔵されているので、あとはキー入力のためのスイッチを接続すれば最低限の USB キーボードが成立します。

そしてオープンハードウェアとして開発されているキーボードはかなりのものが qmk_firmware (https://github.com/qmk/qmk_firmware) というファームウェアで制御されています。qmk は度重なる機能追加で入り組んだソースになっていて、一度にすべてを理解するのはすこし難しいかもしれません。しかし、キー配列やスイッチ・マトリックスのスキャン処理はカスタマイズし易いようになってます。

キーボードのキットや部品を販売しているネットショップも海外頼りだったところが、国内からも出てきていて、キーボードを自作する敷居はどんどん下がっています。

4. どんなキーボードを自作できるのか？

図 4-1 に筆者の現在のキーボードを示します。

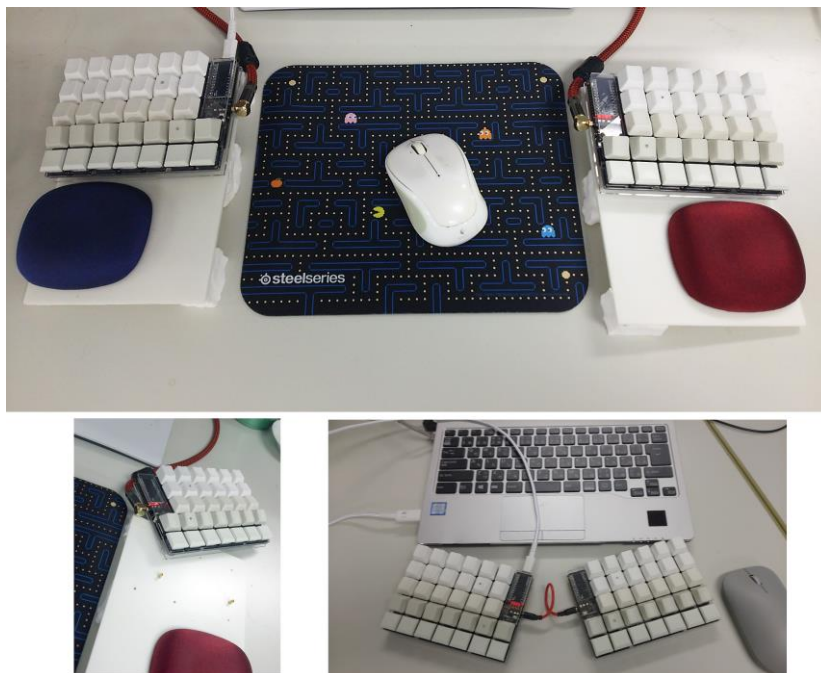


図 4-1 筆者のキーボード(2018 年 7 月)

筆者はなぜこのようなキーボードを作ったのでしょうか？

表 4-1 に要望と要件を示します。

表 4-1 要望と要件

項番	要望	要件
1	肩こりを低減したい	ホームポジションからマウスまで手の移動距離が短いこと。
2		手首の角度をなくし前腕のストレスをなくすこと。
3		Shift、Ctrl は親指で押すこと。小指を酷使しないこと。
4	入力効率を向上したい	QWERTY 配列でなく Colemak 配列とし指移動を減らすこと。
5		記号入力に英語配列に近いこと。 ただし、PC の OS は日本語キーボード配列の設定であること。 例:Shift+' で"が入力される
6		Shift、Ctrl は親指で押すこと。ホームポジションで打てること。
7		自席デスクトップ PC の他に、会議室でノート PC に繋いで使用できること。

このキーボードは万人にお勧めできるものではありませんが、一般的なキーボードに比べ筆者の身体や開発環境からくる要望をより良く実現しています。

要望は昔から変わりませんが、最初からこの様なキーボードを目指していたわけではありません。また、これが最後のキーボードとも思っていないです。

筆者が最初に格子配列キーボードを見たときには何とも言えない気持ち悪さを感じました。QWERTY の他に Dvorak というキー配列があるのは知っていましたが、Colemak 配列など聞いたこともありませんでした。親指シフトキーボードというものがあるのは知っていましたが、Ctrl や Alt まで親指で操作できるのか、それで入力効率が上がるのか確信は持てませんでした。

このキーボードは、どうしたら要望が実現できるのか PDCA サイクル(図 4-2)を回し幾つもキーボードを試作し、DevOps(図 4-3)を実践する毎日を過ごしてきた結果なのです。

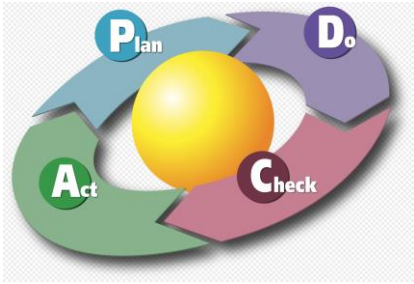


図 4-2 PDCA サイクル

https://ja.wikipedia.org/wiki/PDCA_サイクル#/media/File:PDCA_Cycle.svg より引用

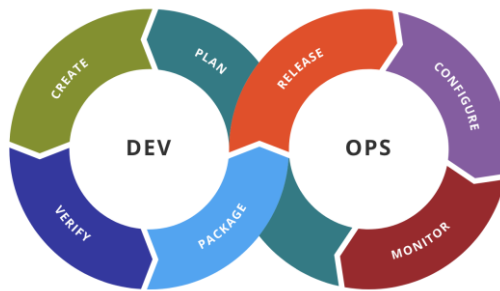


図 4-3 DevOps

<https://upload.wikimedia.org/wikipedia/commons/0/05/Devops-toolchain.svg> より引用

図 4-4 に筆者のキーボード製作の軌跡の一端を示します。

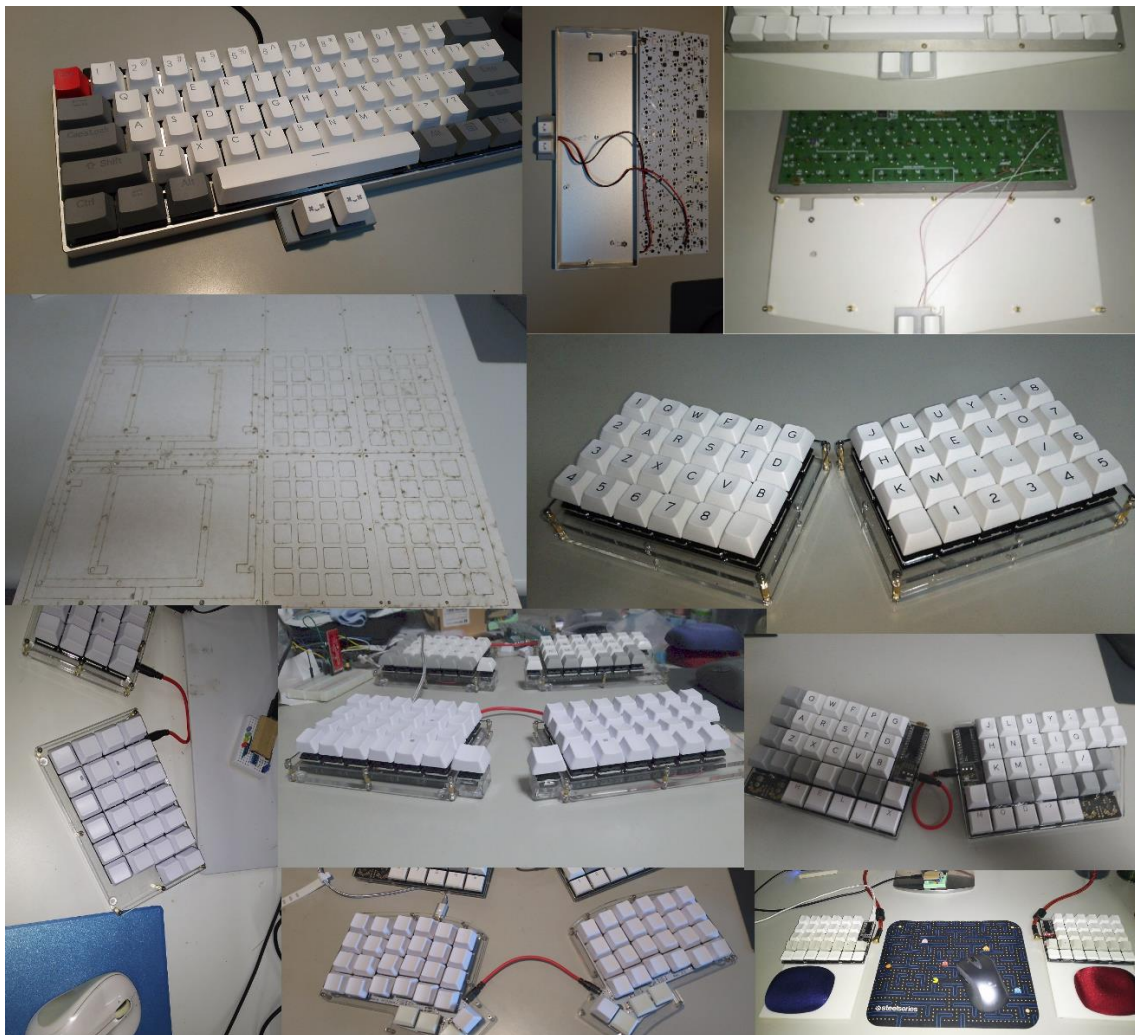


図 4-4 筆者のキーボード製作の軌跡

5. まとめ

キーボードを自作する敷居は下がっています。国内のネットショップも出てきました。

毎日使うキーボードを改善していくなかで PDCA や DevOps を実践しましょう。自分が開発者でありながら顧客であり、改善結果は業務の効率化という実利に繋がっています。ただの演習とはそこが違います。

ソフトウェアの世界に留まらず、他の分野・新しいものに挑戦し、結果を評価し、つぎの挑戦に繋げる。そういった日頃の姿勢が実務にも良い影響を与えてくれることを実感しています。

当日のセッションでは各要素についてより深く掘り下げて説明させていただきます。