

DIY 的 IoT サービスに向けた運用・維持・管理 SaaS のサーバレスによる実装 — motch: 誰でも簡単に導入できる IoT 管理 SaaS —

1. 背景

IoT の普及に伴うマイコンボードの低価格化や文献の充実により、一般消費者が DIY 的に IoT システムを構築できるようになった。今後より IoT が拡大していくためには、DIY 的な IoT の活用が現場で働く人々に波及し、現場の人々の問題意識が自らの手による DIY 的な IoT システムの構築と運用、すなわち DevOps に繋がるのが重要である。現場の人々による DIY 的な IoT システムの構築は文献を参照することで比較的容易に実現することができるが、運用に関する知見はまだ蓄積されておらず、文献を参照しても得られる情報が少ない。さらに既存技術の IoT 向け PaaS、監視 SaaS、IoT 統合プラットフォームは、仕様がまちまちな IoT システムにそのまま導入して監視をすることができるものではない。このため、現状では現場の人々が DevOps のうち「Ops」の部分の担い手となるのが難しい。

2. 目的

本プロジェクトでは、DIY 的な IoT システムの管理を容易にすることを目的とした。どのような IoT システムも管理を必要としているため、IoT システムの管理の部分を SaaS として提供することにより、現場の人々によって DIY 的に作られた IoT システムに管理の機能を容易に追加することができ、現場の人々による IoT システムの DevOps を助けることができる。

3. 開発の内容

本プロジェクトでは、DIY 的な IoT システムに向けた運用・維持・管理 SaaS 「motch」を開発した。motch の全体像を図 1 に示す。

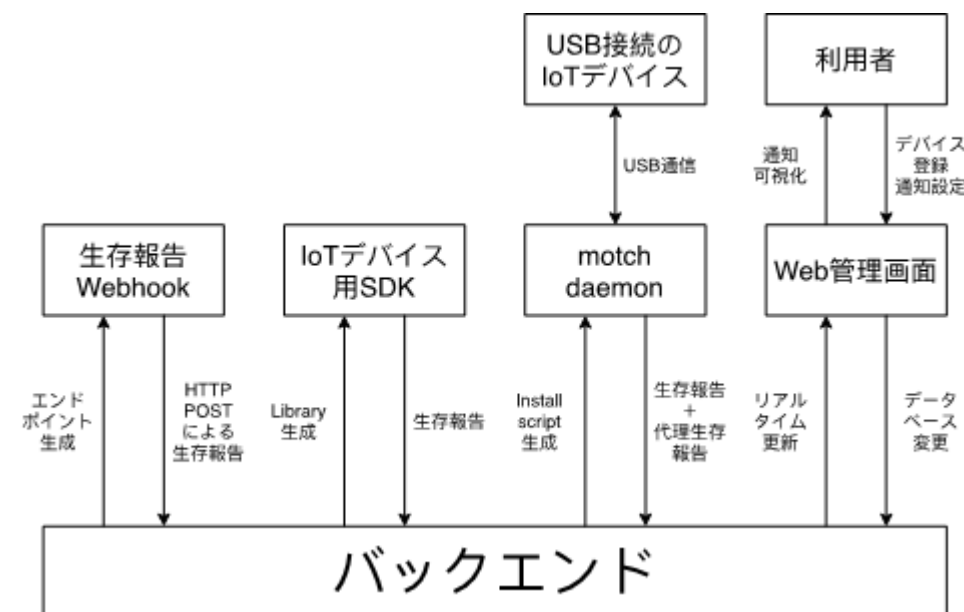
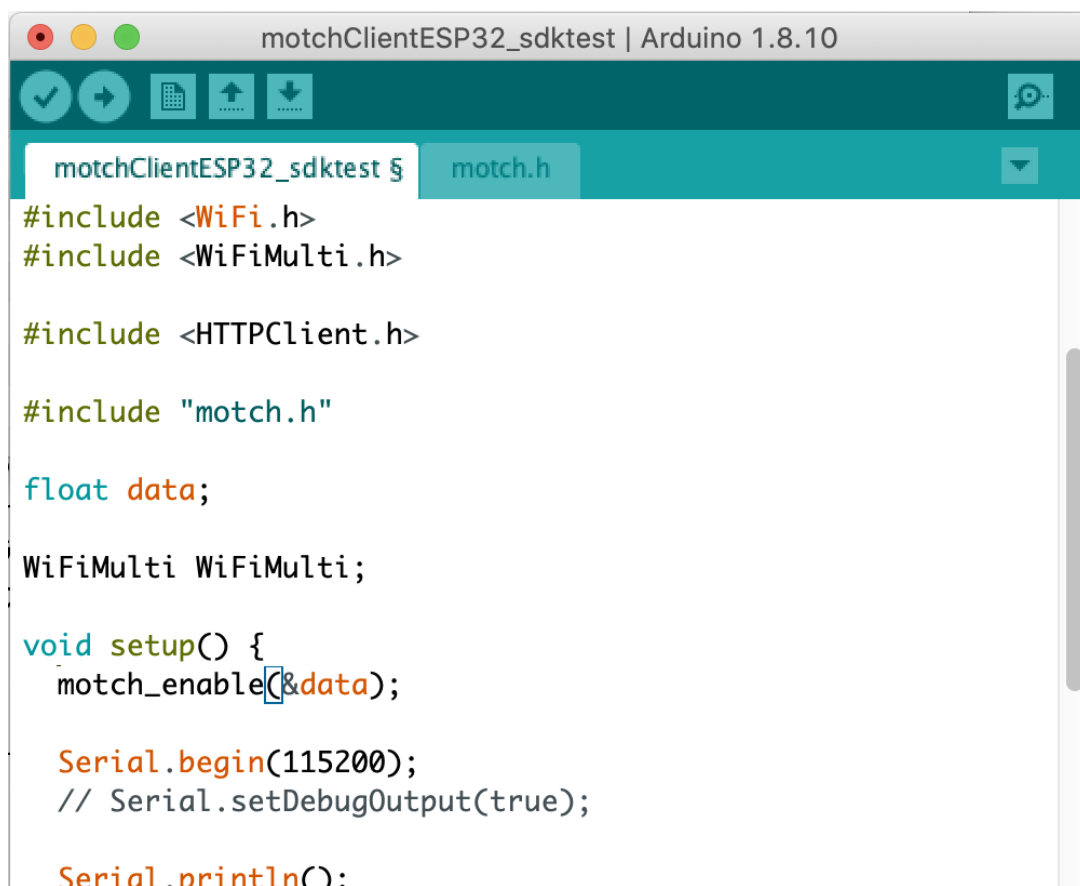


図 1. motch の全体像

motch は DIY 的に IoT システムを構築した人々に対し、IoT システムに管理の機能を手軽に導入するための機能を提供する。motch の構成要素は管理サポート機能、バックエンドの 2 つである。

管理サポート機能は、IoT デバイスからの生存報告をサポートする生存報告 Webhook、motch SDK、motch daemon の 3 つの機能と、IoT デバイスを管理するための Web マネージャで構成される。利用者は motch を用いて IoT システムに管理機能を導入する場合、IoT システムの構成に応じて生存報告サポート機能の中から必要な機能を選択し、IoT デバイスに導入する。生存報告 Webhook はデバイスが定期的にデータを送信する IoT システムにおいて、データを受信するサーバなどに受信時処理の一つとして POST リクエストを送信させることで生存報告を行う。motch SDK はデバイスが定期的にデータを送信しない IoT システムなどにおいて、IoT デバイスのプログラムに 2 行追加するだけで生存報告機能を実装することができる。motch SDK を用いてデバイスのプログラムに生存報告を実装している様子を図 2 に示す。



```
motchClientESP32_sdktest § motch.h
#include <WiFi.h>
#include <WiFiMulti.h>

#include <HTTPClient.h>

#include "motch.h"

float data;

WiFiMulti WiFiMulti;

void setup() {
  motch_enable(&data);

  Serial.begin(115200);
  // Serial.setDebugOutput(true);

  Serial.println();
```

図 2. motch SDK を利用した生存報告の実装

motch daemon は自力で生存報告を行うことができないデバイスの生存報告を代理で行う機能をソフトウェア的に提供する。Web マネージャは、生存報告サポート機能によって生存報告をしている IoT デバイスの稼働状況を可視化するだけでなく、IoT デバイスの情報設定やセンサデータを元にした通知の設定なども行うことができる。motch によってデバイスの稼働状況を閲覧している様子を図 3 に示す。



図 3. Web マネージャのデバイス一覧

バックエンドはリクエスト処理, UI のリアルタイム更新, 死活監視, デバイス情報の管理と SDK の生成, SDK とクラウドリソースの CI/CD の 5 つの機能で構成される。これらの機能により, IoT デバイスの生存報告や死活監視, Web マネージャの動作, および SDK などの開発基盤が支えられている。バックエンドのアーキテクチャ図を図 4 に示す。バックエンドの実装にはサーバレスアーキテクチャを用いることで, 運用コストの低減と高いスケーラビリティを実現している。

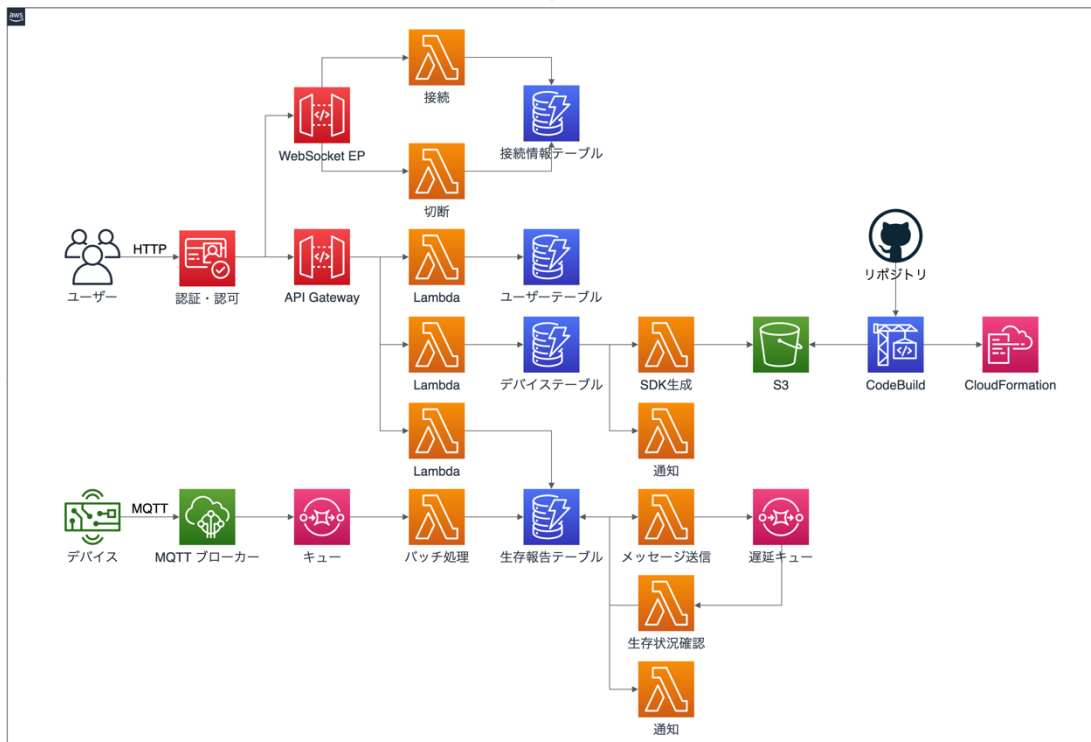


図 4. バックエンドのアーキテクチャ図

4. 従来の技術(または機能)との相違

開発した motch には従来技術との相違点が 2 点ある。一つは IoT デバイスに手軽な生存報告を提供することに主軸を置いている点である。既存技術は、現場の人々や一般消費者といった IoT システムの運用に長けていない人々に、要求を元に制作したシステムの運用を容易に実現するための機能を提供できていない。それに対し motch は多様な IoT デバイスの管理・監視に特化したシンプルな機能を提供する。これにより、既存技術で課題であった、管理システムを既存の IoT システムに大きな変更を加えずに導入できないという点を解決した。また、マルチベンダな IoT システムに手軽に監視機能を追加できる SaaS は唯一のものである。

もう一つは、運用コストの低減と高いスケーラビリティを実現するためにサーバレスアーキテクチャを採用した点である。開発したバックエンドはサーバレスアーキテクチャの採用により自動的にスケールアウトするため、利用者の増加に応じて柔軟にスケーリングすることができる。またサーバレスアーキテクチャを採用して各サービスを従量課金で利用することに強く拘ったため、ユーザ登録数が少ない SaaS 公開直後などは非常に低いコストで運用することができる。例えば、デバイスがたった 1 台しか登録されていないという状況において、同様の SaaS を AWS EC2 などの仮想サーバ上で公開すれば月数千円の固定費が発生するのに対し、motch のバックエンドは月たった数円で運用することができる。

5. 期待される効果

motch の導入により、あらゆる IoT システムの運用を容易にすることができる。一般消費者や現場の人々による DIY 的な IoT の活用がさらに進み、IoT による現場の改善やホームオートメーションの導入が今後さらに広まっていくと期待される。

6. 普及(または活用)の見通し

引き続き、motch を研究プロジェクトおよび趣味における IoT 開発に役立てる予定である。また、motch を SaaS として公開する予定である。クラウドリソースの利用料を賄うため、多くのデバイスを登録するユーザ向けにサブスクリプション制を設ける。需要に応じて、収益化も視野に入れて開発を継続する予定である。

7. クリエータ名(所属)

永野 元基(大阪大学工学部)

新井 悠介(大阪大学工学部)

(参考)関連 URL

motch: <https://www.motch.info/>