

修士論文

題目

IoT機器からの通知に基づいた機器監視サービスの開発

学籍番号・氏名

15006 宮坂 虹櫻

指導教員

横山 輝明

提出日

2017年2月13日

神戸情報大学院大学
情報技術研究科 情報システム専攻

目 次

第 1 章 はじめに	1
1.1 研究の背景	1
1.2 IoT サービスの維持における課題	1
1.3 研究の目的	2
1.4 本論文の構成	2
第 2 章 IoT サービスの提供における問題	3
2.1 IoT サービスの構造と事例	3
2.2 IoT サービス開発の事例	4
2.2.1 株式会社ルナネクサスでの事例	4
2.2.2 岡本商店街における事例	5
2.3 IoT サービスの提供者の課題	6
第 3 章 既存の監視手法	7
3.1 サーバからの問い合わせによる監視	7
3.2 監視対象機器からの通知による監視	8
3.3 株式会社 SORACOM が機能として提供している監視	9
3.4 VPN の利用	9
3.5 既存手法のまとめ	9
第 4 章 IoT 機器からの通知に基づく機器監視サービスの提案	11
4.1 既存手法の問題点	11
4.2 IoT 機器監視・管理サービスの提案	12
第 5 章 機器監視サービスの実装	14
5.1 要件と機能の抽出	14
5.2 提案サービスの構成	14
5.3 エージェントプログラムの設計と実装	15
5.4 エージェントプログラム用インターフェースの設計	16

5.5 エージェントプログラムとエージェントプログラム用インターフェース間の通信	18
5.6 機器状態データベースの設計	19
5.7 機器情報データベースの設計	20
5.8 可視化アプリケーションの設計	21
5.9 サービス利用のイメージ	21
5.9.1 機器の追加	21
5.9.2 機器の削除	21
5.10 実装	22
第 6 章 機器監視サービスの動作テスト	23
6.1 検証目的	23
6.2 検証方法	23
6.3 検証結果	23
6.4 考察	25
第 7 章 考察	26
第 8 章 おわりに	28
謝辞	29
参考文献	30
添付資料 1 岡本商店街での事例	31
添付資料 2 ソースコード	35

図 目 次

2.1 IoT サービスの構造図	3
2.2 株式会社ルナネクサス サービスイメージ図	4
2.3 岡本商店街人流観測 構成図	5
4.1 サービス構成図	12
4.2 サービス構成図	13
5.1 サービス構成図	15
5.2 エージェントプログラムの動作	16
5.3 エージェントプログラム用インターフェースの動作（機器から登録用トークンが送られてきた場合）	17
5.4 エージェントプログラム用インターフェースの動作（機器からの状態の通知があった際）	18
5.5 機器 ID の取得/発行・登録時の通信	19
5.6 機器状態の通知/監視時の通信	19
5.7 ユーザテーブルと機器情報テーブルの関連	20
6.1 RaspberryPi2 と IntelEdison	24
8.1 岡本商店街人流観測 構成図	32
8.2 岡本商店街人流観測 使用した機器	33
8.3 岡本商店街人流観測可視化アプリケーションスクリーンショット	34

表 目 次

3.1 既存手法の比較表	10
5.1 機器状態管理テーブル	20
5.2 ユーザテーブル (Users)	20
5.3 機器情報テーブル (Devices)	21

内容梗概

近年，IoT が注目を集めている。IoT とは，コンピュータをさまざまなモノに取り付け，利便性の向上を図る概念である。近年の半導体技術の進歩により，コンピュータが安価・小型になったこと，インターネットへの通信が様々な場所で安価に行えるようになったことにより，安価に実現可能となった。

それらのモノが連携して提供するサービスは IoT サービスと呼ばれ，より生活に身近なサービスの登場が期待されている。IoT サービスは，IoT 機器とサーバがインターネットを介して通信し合うことで，成り立っている。IoT 機器は，モノにコンピュータが取り付けられた機器で，周囲の状況を検知，または，周囲へ働きかける機能を持つ。サーバは，IoT 機器からの情報を蓄積・分析し，IoT 機器へ指示を送るか，ユーザへ分析結果を表示する機能を持つ。これら IoT 機器とサーバが連携することで，IoT サービスは利便性をユーザへ提供している。

IoT サービスを円滑に提供するには，IoT 機器とサーバの連携を正常に維持しなければならない。そのため，IoT 機器の動作状態や通信状態の監視が重要となる。IoT 機器の監視は，接続されるネットワークが多様であることから，通知型の監視を用いる必要がある。そのため，多量に存在する IoT 機器へ個別に設定をする必要がある。また，多量の IoT 機器を監視サーバへ登録する必要もある。これら設定は，IoT 機器の交換や追加の際にも発生するため，IoT サービスの開発・運用の負担となっている。

本研究では，これら負担が，IoT 機器と監視サーバに分散して設定が存在することが原因だと考えた。そこで，分散して存在する設定をサーバにて一元的に管理し，IoT 機器への個別の設定，監視サービスへの登録作業を省力化することを提案する。サーバから IoT 機器へ個別の ID を付与し，機器への設定を簡略化し，ID を付与した時点で機器を登録することで，登録を自動化する。機器は，個別の ID に紐付いた URL へ https にて通知することで，機器の監視を可能にした。

第1章 はじめに

1.1 研究の背景

近年，IoT が注目を集めている。IoT とは，様々なモノがインターネットにつながり，相互に情報をやり取りすることで，利便性の向上を図る概念である。近年の半導体技術の進歩により，コンピュータが安価・小型になったこと，インターネットへの通信が様々な場所で安価に行えるようになったことにより，注目が集まっている。IoT の具体的な例としては，建設重機の盗難防止，プリンタのトナー発注自動化，体温や脈拍等の収集・可視化等が挙げられる。

IoT サービスとは，IoT による利便性を顧客に提供するサービスであり，より生活に身近なサービスの登場が期待されている。上記の例では，プリンタのトナー発注自動化が IoT サービスにあたる。IoT サービスは，IoT 機器とサーバがインターネットを介して通信し合うことで成り立っている。IoT 機器は，モノにコンピュータが取り付けられた物で，周囲の状況を検知，または，周囲へ働きかける機能を持つ。サーバは，IoT 機器からの情報を蓄積・分析し，IoT 機器へ指示を送るか，ユーザへ分析結果を表示する機能を持つ。これら IoT 機器とサーバが連携することで，IoT サービスは利便性をユーザへ提供している。

このような IoT サービスを円滑に提供するためには，IoT 機器・インターネットへの接続・サーバの状態を監視し，必要に応じて，機器の交換等を行う必要がある。

1.2 IoT サービスの維持における課題

しかし，IoT 機器の監視は，技術的課題から既存手法の適応が難しい。既存手法は，機器の監視のためには，監視サーバと機器の両方に設定が必要となる。IoT 機器が多量に使用されること，IoT 機器の追加・撤去・交換は頻繁にあることから，監視サーバへの登録や IoT 機器への個別の設定が，大きな負担となっている。

IoT サービスの提供者として，太陽光発電発電に係る機器の監視サービスを提供している株式会社ルナネクサスが挙げられる。株式会社ルナネクサスとは，大阪にある組み込み機器メーカーである。株式会社ルナネクサスでは，太陽光発電事業を展開している事業主に対し，発電に係る機器の状態や，発電量等を可視化できるサービスを展開している。しかし，IoT 機器の数に増減があることや，故障などによる交換により次のような事が問題となっている。

- 各 IoT 機器の状態を監視するために、多数の IoT 機器へ個別の設定をしなければならないこと
- 増減や交換の度に、機器監視システムへの登録をしなければならないこと
- 交換の為に現地に行くも、類似の機器が多数存在し、外観から交換対象機器がわかりづらいこと

1.3 研究の目的

そこで、本研究では、IoT 機器へ個別の ID を付与しなければならない負担、監視サービスに登録を行わなければいけない負担を軽減するために、監視サーバから IoT 機器に対して ID を配布し、監視サーバにて ID の一元的な管理を提案し、システムを作成する。サービス管理者は、機器の追加の際に、監視サービスから登録用 ID と URL を取得し、全ての機器に対し同一の ID と URL を設定し、機器を起動させる。機器側では、登録用 ID と URL から、監視サービスにアクセスし、個別の ID の発行を受ける。監視サービス側では、アクセスのあった順に重複の無い ID を発行し、登録を行う。管理者は、機器を起動した順に ID の印刷、貼り付けを行う。

前項に挙げた IoT 機器の監視の為の管理が負担である問題は、機器監視サーバと各 IoT 機器に設定が分散しているのが原因であると分析し、機器監視サーバでの設定の一元管理と、各 IoT 機器への個別の設定の簡略化により、負担を軽減することを目的とする。

1.4 本論文の構成

第 2 章では、IoT サービスの維持に関する背景と、IoT 機器の監視・管理に関する問題を述べる。第 3 章では、第 2 章で述べた問題を分析し、IoT 機器の監視・管理はどうあるべきか提案を述べる。第 4 章では、IoT 機器監視・管理サービスの実装の詳細について述べる。第 5 章では、実験により IoT 機器監視・管理サービスがもたらす効果を検証し、考察を述べる。第 6 章では、本研究に関する評価について述べる。第 7 章では、本研究を通して得られた知見や今後の課題について述べる。

第2章 IoT サービスの提供における問題

2.1 IoT サービスの構造と事例

IoT とは、「モノのインターネット」とも呼ばれる、様々なモノがインターネットを介して通信し合うことで自動化を図る概念である。IoT サービスとは、IoT による利便性を顧客へ提供するものである。IoT サービスは、図のように IoT 機器、ネットワーク（インターネット）、サーバから構成されている。

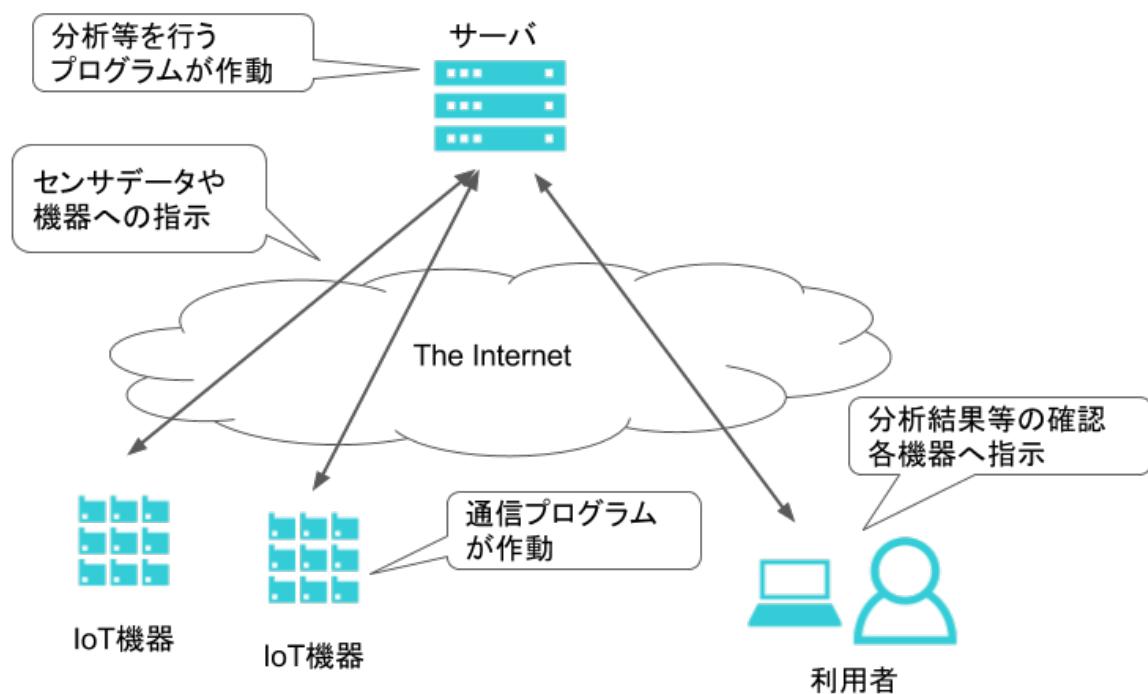


図 2.1: IoT サービスの構造図

IoT 機器は、RaspberryPi や Edison 等の小型コンピュータや組み込み機器が使用され、周辺環境を検知する他、周辺環境への働きかけを行う。IoT 機器では、通信プログラムが作動しており、インターネットを介して、検知した周辺環境に関する情報や、IoT 機器への指示をやり取りする。IoT 機器は、Wifi や有線接続等を用いて、インターネットへ接続された携帯電話網や社内 LAN 等のネットワークに接続される。

サーバは、インターネット上に設置され、蓄積・分析・可視化等を行うプログラムが作動している。このプログラムは、IoT 機器から送られてきた情報を蓄積・分析し、必要に応じて IoT 機器へ指示を送る他、分析結果を利用者へ可視化する。利用者は、サーバへインターネットを介して接続し、分析結果の閲覧・機器への指

示を行う。

2.2 IoT サービス開発の事例

2.2.1 株式会社ルナネクサスでの事例

株式会社ルナネクサスでは、太陽光発電事業を展開している事業主に対し、発電に係る機器の状態や、発電量等を可視化できるサービスを展開している。太陽光発電は、太陽光パネルと発電した電力を送電する為の「パワーコンディショナ」と呼ばれる機器で成り立っている。

株式会社ルナネクサスでは、そのパワーコンディショナに独自に開発した IoT 機器を取り付け、発電量や発電機器の異常などを収集する。収集したデータは、SORACOM Air という携帯電話網を利用したインターネット接続サービスを使用してインターネット上にあるサーバへ送信される。サーバ上では、各 IoT 機器から送られてきた情報を蓄積し、可視化する。これによって、発電事業を展開している事業主に、各発電所まで行かなくても発電量や発電機器の異常等を確認することができる、という利便性を提供している。図 2.2 は株式会社ルナネクサスの IoT サービスイメージである。太陽光発電所は僻地にあることが多いので、利用できるネット

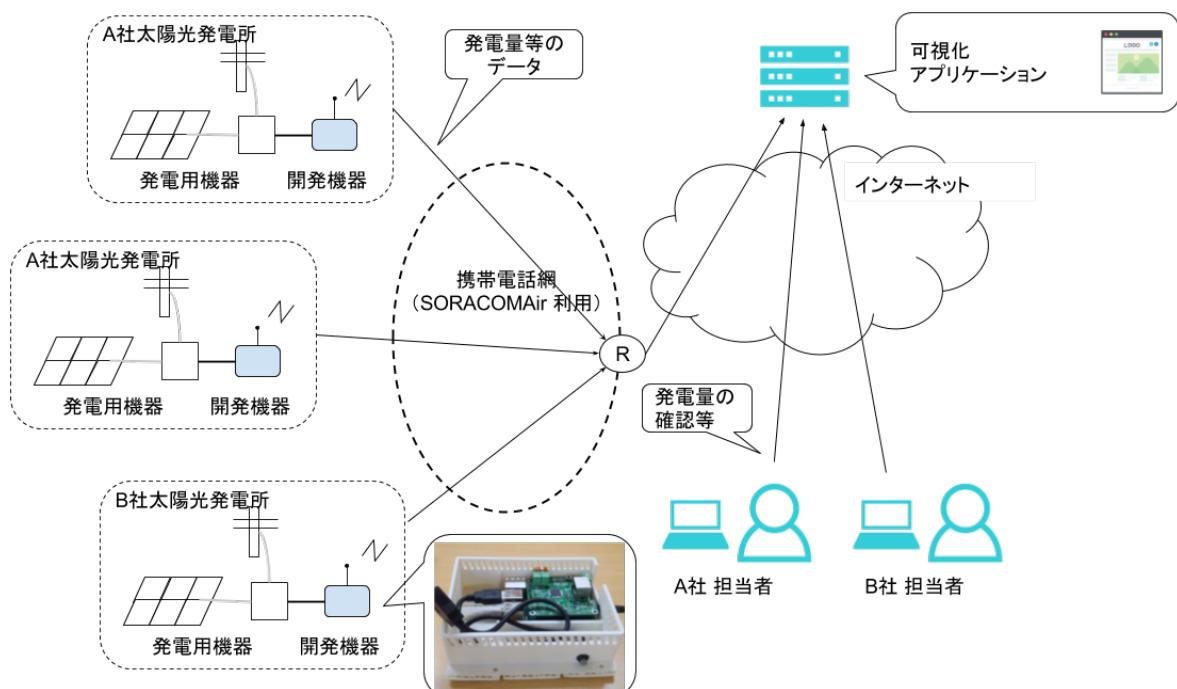


図 2.2: 株式会社ルナネクサス サービスイメージ図

ワークが無いことが多い。このことから、インターネットへの接続に SORACOM Air を選択した。

しかし、SORACOM Air 回線は、プライベートアドレスを使用しており、IoT 機器からの通信は通過するものの、インターネット側から Ping などによる確認を行うことが出来ない。そのため、各 IoT 機器の監視とメンテナンスの為に、VPN を利用し、定期的に VPN 越しにログインすることで監視を行っている。VPN 越し

にログインするためには、各 IoT 機器に VPN クライアントを導入し、新たに VPN サーバを立ち上げる必要があった。また、手動で各機器へログインすることは、手間である。

2.2.2 岡本商店街における事例

岡本商店街とは、神戸市東灘区にある阪急岡本駅と JR 摂津本山駅の間に位置する商店街のことである。商店街の方に人流を可視化する IoT サービスを提供し商店街の活性化に役立てるといった趣旨で、2016 年 2 月 7 日から 2016 年 3 月 14 日まで観測を行った。人流観測とは、各地点から各地点迄をある時に移動した人数を観測するものである。今回は、ある地点を通過した人物が以前どの地点を通過していたのかを観測したかった為、携帯電話についている WiFi 機能が送出する電波を利用して観測を行うこととした。

観測・分析・可視化を行うシステムの構成としては図 8.1 の様になる。各店舗に設置させてもらった観測機器は、店舗に敷設されたネットワーク回線を通して、インターネットへと繋がっている。各観測機器は、観測したデータをインターネット上のサーバへと送信し、そのサーバの上で蓄積・分析・可視化を行う形となっている。

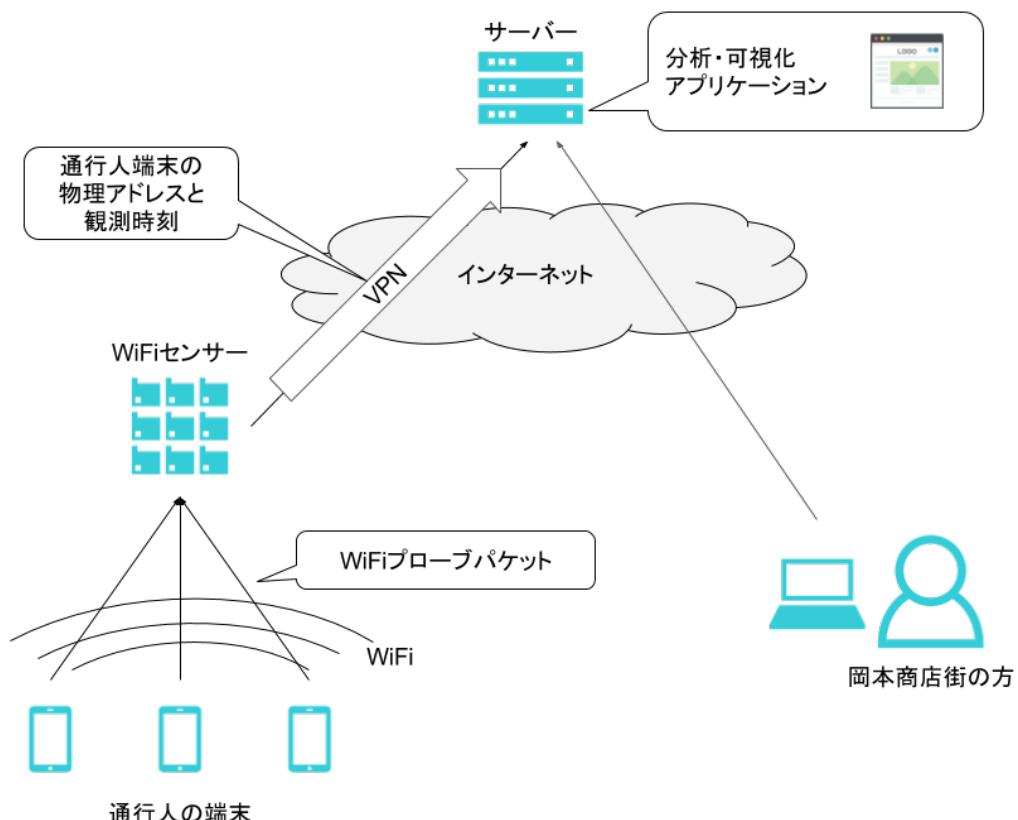


図 2.3: 岡本商店街人流観測 構成図

実験の際に、次の様な問題や負担があった。

- 電源が抜けている・ネットワークが切断されているといった要因で、観測できていないことがあった。
- 上記要因にてトラブルに気づくことが遅れた。

これらから、IoT 機器の監視が必要と考え、VPN 越しにログインすることで、確認を行うこととした。しかし、各観測機器に VPN を介してログインすることは手間であり、負担であった。また、実験では観測機器の数が 5 台のみであったが、大規模に観測を行うことを考えた場合、手動での確認はとても大きな負担となる。そのため、効率化を図る必要があったと感じる。

2.3 IoT サービスの提供者の課題

これら事例と実験から IoT サービスの円滑な提供において、開発は容易だが、IoT サービスの構造を維持するため、IoT 機器の監視・管理の必要があることが分かった。しかし、IoT 機器の監視・管理には、次のような課題がある。

- IoT 機器が多量である
そのため、各機器へ設定することは大きな負担となる。また、監視サーバへの登録も負担となる。
- IoT 機器が接続されるネットワークが多様である
そのため、既存の問い合わせによる監視手法は適応できない。しかし、通知による監視を行うには、各機器へ設定する必要がある。
- IoT 機器の追加や交換が頻繁にある
そのため、監視サーバへの登録作業を頻繁に行う必要がある。
- 外観が似通っている
そのため、交換のため現地に行くも、交換すべき機器がどれなのか判別がつかない。

私は、その中でも、監視サーバへの登録の負担と IoT 機器への設定の負担について取り上げ分析することにした。IoT 機器の監視のためには、IoT 機器が接続されるネットワークの多様性から、IoT 機器からの通知による監視手法を取らざる終えない。しかし、そのためには IoT 機器と監視サーバに対し、識別子 (ID) を設定する必要がある。IoT 機器に対し、固有の識別子を与え、監視サーバに識別子を伝えることで、監視サーバはどの機器からの通信なのか見分けることができる。また、監視サーバに対し、予め識別子を設定しておくことで、監視対象とする機器以外からの不正なアクセスを遮断することができる。よって、IoT 機器へ識別子を設定し、監視サーバに対して登録を行う必要がある。しかし、IoT サービスでは、IoT 機器が多量に使用されることや、頻繁な交換・追加・撤去がありうるため、これら設定や、設定の変更は大きな負担となっている。

第3章 既存の監視手法

岡本商店街での実験や、株式会社ルナネクサスへの聞き取りから、IoT サービスの構成上、IoT 機器の監視には次のような要件を満たしている必要があることが分かった。

- IoT 機器の接続されるネットワークが、プライベートアドレスを使用していても、監視可能であること
IoT 機器が接続されるネットワークを予め予測することは難しい。そのため、IoT 機器にプライベートアドレスが割り振られる可能性がある。そのような状況でも監視することができる必要がある。
- ネットワークが違っていても、一つの画面で確認できること
- IoT サービスの変更が不要であること
サービスの変更是開発者への負担となるため、サービスへの変更を避けたい。
- 監視サーバを立てる必要がないこと
新たに監視サーバを立てるとは、設定や構築作業が必要となるため、開発者の負担となる。そのため、新規に監視サーバを立てる必要が無いことが求められる。

3.1 サーバからの問い合わせによる監視

従来から機器監視に用いられてきた手法として、定期的に機器監視サーバから対象機器に状態を問い合わせる手法がある。機器監視サーバから、監視対象機器上のエージェントプログラムに、現在の状態を問い合わせる事で機器の監視を実現している。

代表的なものとして、次のような物がある。

- Ping
- Ping とは、ICMP メッセージをやり取りするプログラムである。ICMP とは、InternetControlManagementProtocol の事で、IP ネットワークにおいて、IP パケットの不到達等を通知する為の取り決めである。ICMP には、ICMP Echo Request Echo Reply が定義されており、ICMP Echo Request を受け取った機器は、ICMP Echo Reply を返さなくてはならない。監視を行う機器から、Ping を用いて、ICMP Echo Request を監視対象機器の IP アドレスに送信し、Echo Reply の有無から、機器の存在を確認する事ができる。

- SNMP による問い合わせ

SNMP とは、Simple Network Management Protocol の事で、ネットワークの監視・管理を行うための、取り決めである。SNMP には、SNMP マネージャと SNMP エージェントが存在する。SNMP マネージャは、監視を行う機器上で動作し、SNMP エージェントは、監視される機器の上で動作する。SNMP を利用した機器の状態監視の方法として、SNMP マネージャが SNMP エージェントに問い合わせを行い取得する方法がある。

この手法だと、コンピュータであればほぼすべての機器が対応しているので、手軽に監視できる。しかし、監視する側のコンピュータに、全ての監視対象の IP アドレスを設定する必要があるので、大量に IoT 機器が使用される事を考えると負担となる。また、IoT 機器の接続されるネットワークがプライベートアドレスを利用している場合や、セキュリティの設定によりネットワーク機器にてブロックされてしまう事もある。そのため、この手法を取ることは難しい。

3.2 監視対象機器からの通知による監視

機器監視手法として、定期的に監視対象機器から機器監視サーバへ状態を送信するという手法がある。監視対象機器上のエージェントプログラムが、指定した時間毎に機器監視サーバへ状態を送信することで、機器の監視を実現している。

代表的なものとして、次のようなものがある。

- SNMP Trap による取得

SNMP では、予め SNMP エージェントに設定をしておくことによって、状態の変化を SNMP マネージャへ知らせることができる機能がある。この機能を利用して、SNMP エージェントの状態を取得できる。

- Teregraf と Influxdb による、機器の状態の通知と蓄積

Teregraf とは、監視対象にインストールされるエージェントプログラムで、監視される機器の上で動作する。Influxdb とは、時系列データを格納することに特化したデータベースで、この中では機器の状態を蓄積することに使用され、監視を行う機器の上で動作する。Teregraf は、機器の状態を定期的に取得し Influxdb に送信する。そして、Influxdb は、それを蓄積する。Influxdb に蓄積されたデータは、Influxdb に対応した可視化アプリケーション (Grafana 等) によって可視化され、監視を行うことができる。

この手法だと、前項で述べた手法と比べて、監視を行う機器に対し監視対象の機器の IP アドレス等を登録する作業が不要である。また、プライベートアドレスを利用している場合でも、監視を行うことができる。

しかし、各機器ごとに、通知の送信先を設定しなくてはならない為、負担となる。この手法は、IoT 機器の監視の参考にはなるが、IoT 機器が利用される数を考えると、負担が大きい。また、監視サーバへの登録は行

わなければならず、IoT 機器の多さから負担となる。

3.3 株式会社 SORACOM が機能として提供している監視

ネットワークの提供者がオプションとして提供している監視機能を利用する方法もある。ここでは、SORACOM が提供しているネットワークを利用した場合の監視を取り上げる。

株式会社 SORACOM とは、IoT プラットフォームとして、通信とクラウドへの接続を提供している企業である。通信は MVNO として携帯電話回線網を使用しており、SORACOM 用 SIM を購入し、機器に SIM モジュールを取り付けることで利用できる。株式会社 SORACOM では、各 SIM のアクティベートや通信量設定の変更、クラウドへの接続の設定等を行うための Web インターフェースを提供しており、プラットフォーム利用者はこの Web インターフェースをインターネットを介して利用し、各種設定を行うことができる。その Web インターフェースには、各 SIM の通信量や状態を把握するためのページが存在しており、ここで各 SIM がリンクアップしているかどうかを監視することもできる。

この機能を用いた場合、新たに監視サーバ等を立てる必要が無く、また、インターネットに接続できる環境ならば、どこからでも監視を行うことが可能である。しかし、株式会社 SORACOM が提供しているネットワーク以外のネットワークに接続された機器は監視することができない。

このような独立したサービスは、提供者が監視用サーバをたてなくてはいけないといった負担の軽減になり、また、そのような監視のニーズがある。

3.4 VPN の利用

VPN を利用して、既存の監視手法を使用する方法もある。この場合、機器が利用しているネットワークによらない監視を行うことができるが、監視サーバや機器に対し、IP アドレスを設定することや、各ソフトウェアに設定する事は、改善されない。

3.5 既存手法のまとめ

既存の手法として、サーバからの問い合わせによる監視、監視対象機器からの通知による監視、ネットワークによる監視、VPN の利用といった手法がある。しかし、このような手法では、

1. IoT 機器の接続されるネットワークが、プライベートアドレスを使用していても、監視可能であること
2. ネットワークが違っていても、一つの画面で確認できること
3. IoT サービスの変更が不要であること

4. 監視サーバを立てる必要がないこと

といった事を満たすことが出来ない。

次に示すのは、各要件が満たせるかどうかについての表である。

この表から分かるように、全ての要件を満たした監視手法は無い。また、どの手法をとっても、監視サーバへ

表 3.1: 既存手法の比較表

既存手法 \ 要件番号		1	2	3	4
監視サーバからの問い合わせによる監視	VPN と併用しない場合	X			X
	VPN と併用する場合				X
監視される機器からの通知による監視					X
SORACOM が提供している監視			X		

の登録は不可欠である。この監視サーバへの登録や機器への設定が、機器の数と機器の移動・交換・撤去が頻繁にあることから、大きな負担となっている。これら議論から、IoT 機器の数やに対応した、監視サーバへの登録や IoT 機器への設定が簡単な、監視サービスの存在が求められる。

第4章 IoT 機器からの通知に基づく機器監視サービスの提案

4.1 既存手法の問題点

既存の監視手法として次のような手法がある .

- 監視サーバからの問い合わせによる監視
Ping,SNMP による問い合わせ等がこの手法にあたる .
- 機器からの通知に基づく監視
SNMPTrap による通知がこの手法である .

監視サーバからの問い合わせによる監視を IoT サービスに適応した場合 , 次のような問題がある .

- IoT 機器が接続するネットワークが , プライベートアドレスを使用している場合がある
IoT 機器が接続するネットワークがプライベートアドレスである場合 , 監視サーバから問い合わせを行うことができず , 監視することができない .
- IoT 機器が接続するネットワークにセキュリティの設定が施されている場合がある
特に , インターネットからの特定ポートに対するアクセスをブロックしている場合がある . この場合も , 監視サーバからの問い合わせを行うことが出来ないため , 監視することができない .
- IoT 機器が移動する場合がある
IoT 機器は移動する物に取り付けられる場合がある . この場合 , IoT 機器が接続するネットワークが頻繁にかわるため , 監視サーバが問い合わせを行う宛先を頻繁に更新しなければならない . また , 複数の機器が移動を繰り返す中で , 監視サーバとのやり取りに混乱が生じる .

このため , 監視サーバからの問い合わせによる監視は用いることができない .

一方 , 機器からの通知では , これら機器が接続されるネットワークに起因する問題を解決できる . しかし , 各 IoT 機器に対し個別の ID を設定する必要があり , IoT 機器が大量であることから大きな負担となる . また , どちらの手法を取るにしても , 機器監視サーバへの登録作業は行う必要があり , 負担となる .

4.2 IoT 機器監視・管理サービスの提案

前章で述べたように、IoT サービスの円滑な提供の為には、IoT サービスの構造を維持する必要が有り、そのためには、IoT 機器を監視する必要がある。IoT 機器が接続するネットワークが、プライベートアドレスを利用している等、多様であることから、機器からの通知による監視手法を取る必要がある。しかし、IoT サービスにおいて、IoT 機器は多量に使用されることや、頻繁な交換・追加がありうることから、IoT 機器へ個別の設定をすること、監視サーバへ IoT 機器を登録することが大きな負担となっている。

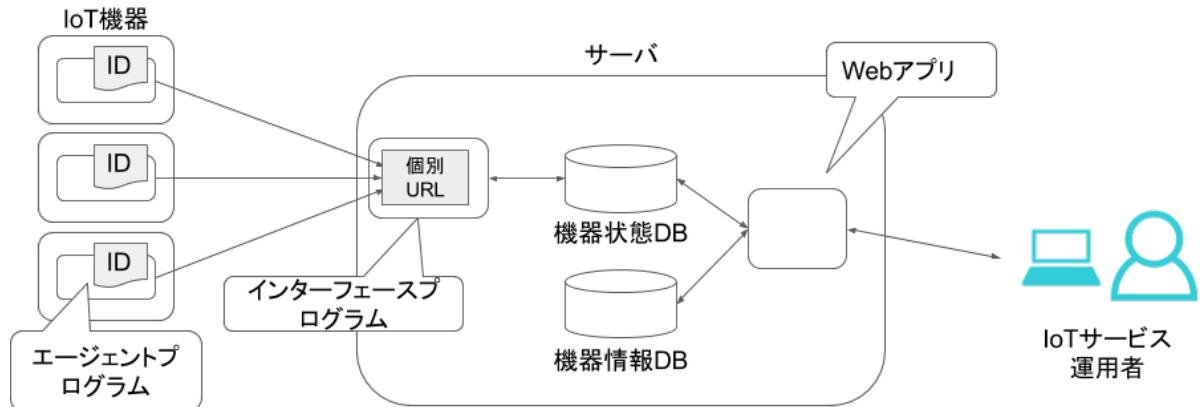


図 4.1: サービス構成図

そこで私は、これら負担は IoT 機器と監視サーバに設定が分散していることが原因と考え、設定を監視サーバにて一元的に管理し、IoT 機器への設定の省力化を提案し、サービスを開発する。このサービスの利用者は、提案するサービスから、機器追加用のトークンを受け取り、各 IoT 機器に対して同一のトークンを設定し、提供するプログラムを作動させる。提供するプログラムは、監視サーバに対して、このトークンを送信し、返答として個別の ID を受け取り、自身に設定する。サーバでは、トークンの有効性を検証し、機器の追加を自動的に行う。図 4.1 は、提案するサービスの構成図である。

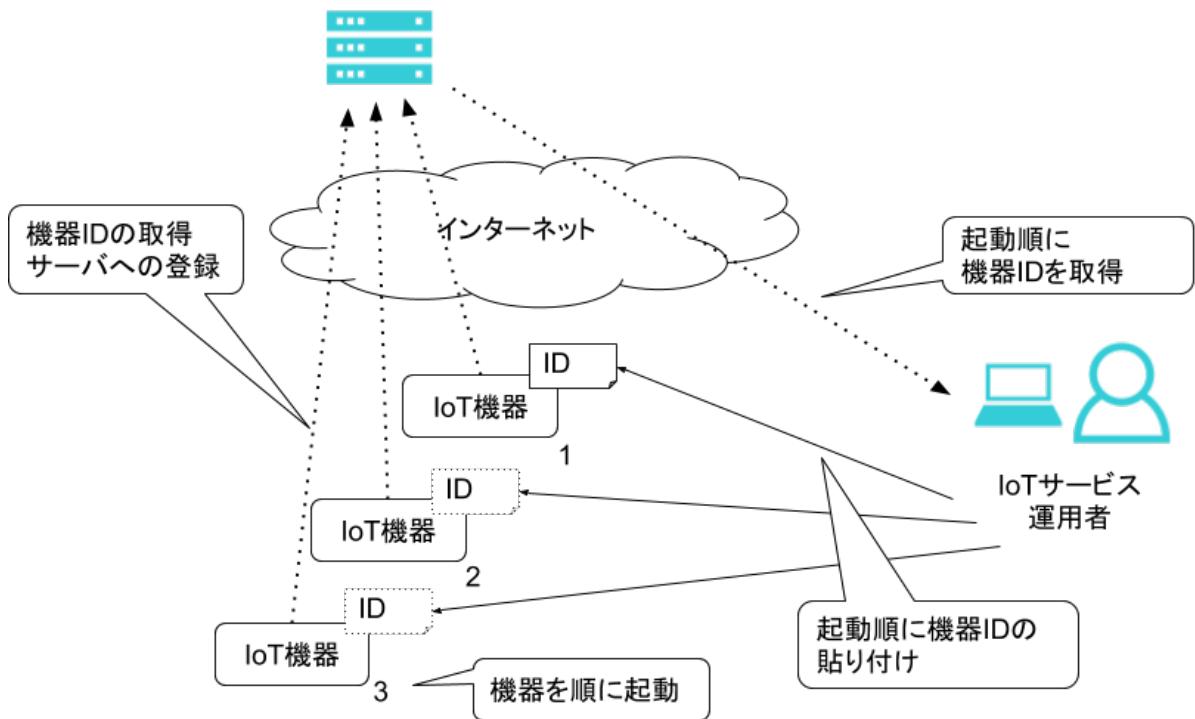


図 4.2: サービス構成図

図 4.2 の様にプログラムを作動させた順に、機器の監視サービスへの登録を確認し、機器に対して ID の貼り付け等を行う。また、IoT 機器上で動作する提供プログラムは、定期的に監視サーバに対しネットワークの状態や稼働状況等を報告することで、監視を行う。

これにより、監視サーバへ各 IoT 機器を登録する負担と、IoT 機器へ個別の設定を行うことを省力化する。さらに、通知型で https を用いた通信により、ネットワークの多様性に対応する。また、サービスとして提供することで、従来では不可欠であった監視サーバの構築作業の負担も解決することができる。

第5章 機器監視サービスの実装

5.1 要件と機能の抽出

以下に要件を整理する。

- IoT 機器への設定が簡略化されること
- IoT 機器が接続されるネットワークの違いによる影響を受けないこと
- 監視サーバ上の論理的な機器と監視対象である物理的な機器との対応がわかりやすいこと

上記要件から、次のような機能を抽出した。

- 機器からの通知による監視情報の収集
- 監視サーバにて機器 ID と URL の対応を管理することで、監視対象機器を特定
- 機器の ID を機器本体に掲示することで、外部からわかりやすくすること

5.2 提案サービスの構成

提案サービスは図 5.1 のように、IoT 機器上で動作するエージェントプログラム、監視サーバ上で動作するエージェントプログラム用インターフェース、機器状態データベース、機器情報データベース、可視化アプリケーションから構成されている。エージェントプログラムは、設定により与えられた機器追加用トークンを監視サーバに送信し、重複無い機器 ID を取得する他、定期的に監視サーバに対して、ネットワークの状態や稼動状態を通知する。エージェントプログラム用インターフェースは、エージェントプログラムから送信された機器追加用トークンの整合性を確認し、機器 ID の発行、データベースへの登録を行う。また、エージェントプログラムから送信されたネットワークの状態や稼動状態をデータベースへ書き込む。機器状態データベースは、機器の状態を蓄積するために使用し、機器情報データベースは、ユーザと機器 ID を結びつけ、機器に関する情報を記憶するために用いる。可視化アプリケーションは、各データベースと連携し、機器追加用トークンの発行や、ユーザ管理、機器状態の可視化を行う。ユーザは可視化アプリケーションへブラウザからアクセスすることで、本サービスを利用する。

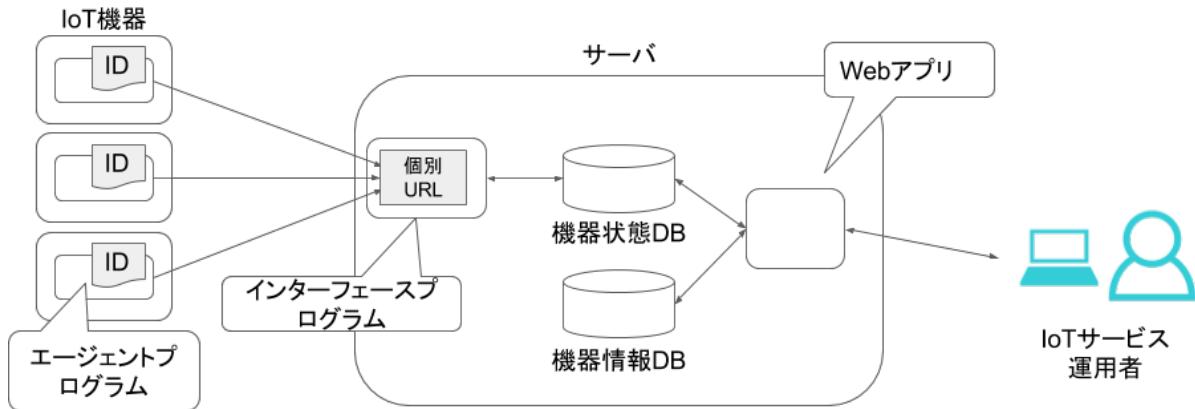


図 5.1: サービス構成図

5.3 エージェントプログラムの設計と実装

エージェントプログラムは、IoT機器上で動作し、IDの取得機能と、定期的な状態の通知機能を持つ。IDの取得機能とは、ユーザから設定されたトークンを監視サーバへ送信し、機器固有のIDを取得・設定を行う機能である。定期的な状態の通知機能とは、1分毎にネットワークの状態、機器の状態を送信する機能である。ネットワークの状態とは、IoT機器から監視サーバへの到達性であり、「監視サーバへ接続できなかった回数」の事とする。また、機器の動作状態とは、エージェントプログラムがサーバへネットワークの状態を通知できているかどうかであり、通知できている場合は「異常なし」と判断する。過去のいつの時点で通信が途切れ、いつの時点で通知不能になったのか、判断するために「過去にサーバへ接続できなかった回数」も合わせて送ることとした。

動作の流れとしては、図のようになる。まず、IoT機器はユーザより設定されたトークンを監視サーバへ送信し、個別のIDを取得する。このIDを自身へ設定しなおし、1分おきに監視サーバのIDと紐付いたURLに対して「過去にサーバへ接続できなかった回数」「現在サーバへ接続出来なかった回数」を送信する。「過去にサーバへ接続できなかった回数」及び「現在サーバへ接続できなかった回数は、サーバへの接続ができた時点で初期状態(0回)に戻る。

実装としては、対象となるIoT機器をRaspberryPiと仮定し開発した。エージェントプログラムは、汎用性を高めるため、シェルスクリプトで作成した。監視サーバへの通信には、セキュリティの設定の有無(ネットワークの多様性)を考慮し、httpsを用いる。また、今回は図にて灰色で示した部分については、実装を行わなかった。

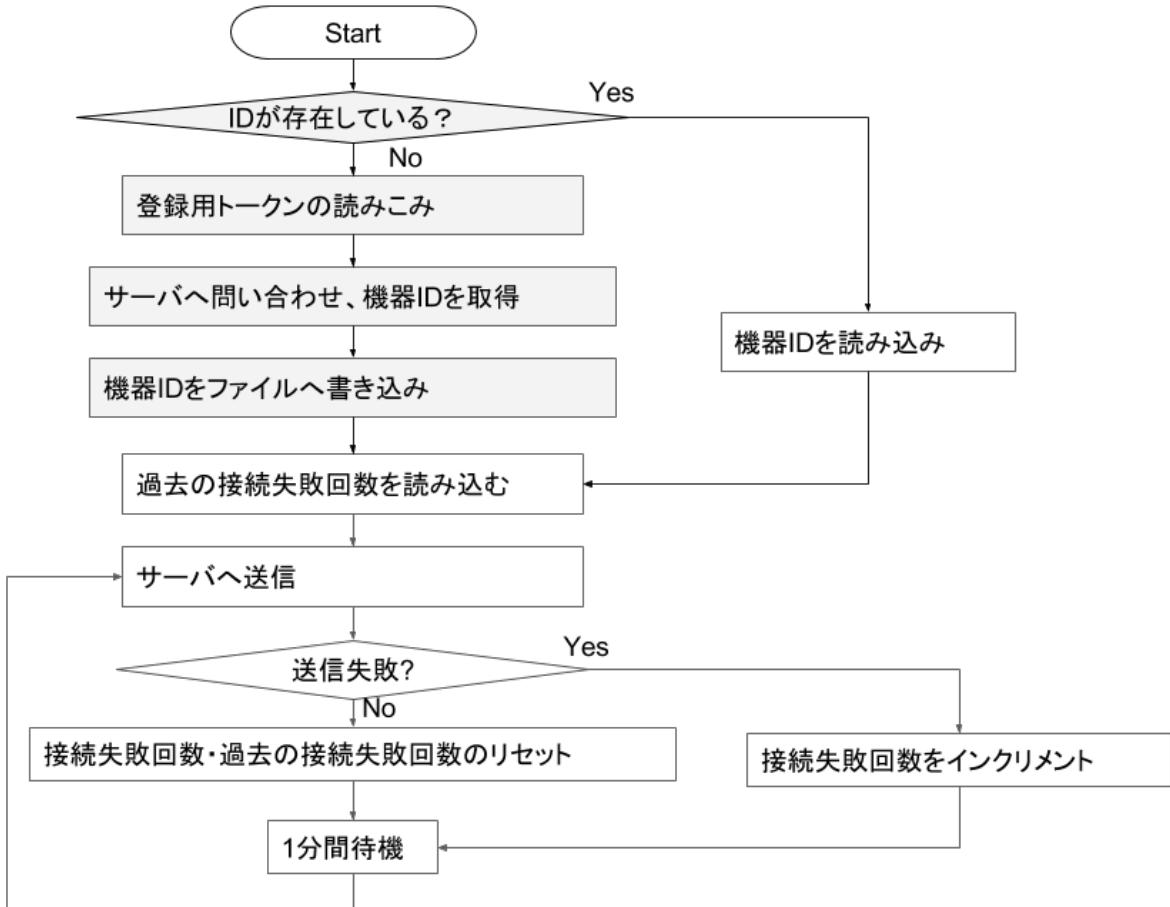


図 5.2: エージェントプログラムの動作

5.4 エージェントプログラム用インターフェースの設計

エージェントプログラム用インターフェースは、監視サーバ上で動作するプログラムで、トークンの検証機能、ID の発行機能、登録機能、機器状態の受け付け機能を持つ。トークンの検証機能とは、IoT 機器から送信されたトークンの整合性を検証する機能である。トークンの整合性とは、トークンに紐付いたユーザが現在機器の追加を許可しているかである。ID の発行機能とは、機器に対し、重複のない ID を発行する機能である。登録機能とは、可視化アプリケーションへ、機器とユーザの関係を登録する機能である。機器状態の受け付け機能とは、IoT 機器から送信された機器状態等から、過去のどの時点でネットワークから切断されたのか等を逆算し、機器状態データベースへ書き込む機能である。

図 5.3 は、エージェントプログラムからトークンが送られてきた場合の動作である。IoT 機器から、トークンが送られてきた場合、インターフェースは次のような動作を行う。

1. トークンの整合性の確認

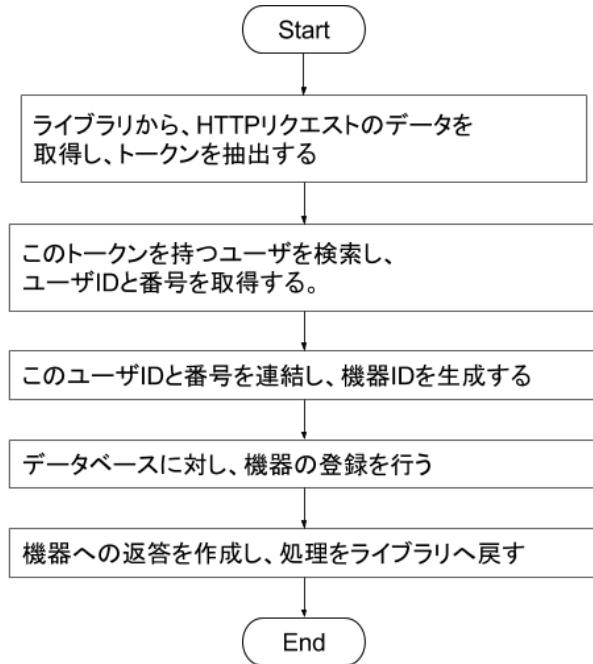


図 5.3: エージェントプログラム用インターフェースの動作（機器から登録用トークンが送られてきた場合）

2. ユーザ情報の取得
3. 機器 ID の生成
4. 機器情報データベースへ関係を格納（登録）
5. 機器に対し、機器 ID を返信

図 5.4 は、エージェントプログラムから機器の状態が通知してきた場合の動作である。

また、IoT 機器から機器の状態が送られてきた場合は、次のような動作を行う。

1. 機器 ID と URL の合致を確認
2. 過去にサーバへ接続できなかった回数と、データベース上の最後の通信の記録から、通知不能になった時刻の推測
3. 現在サーバへ接続出来なかった回数と、現在時刻から、通知可能になった時刻の推測
4. 機器状態データベースに対し、通知不能になった時刻と、通知可能になった時刻、現在時刻の 3 時点に對し、状態の変化を書き込む
5. 現在時刻と受理した旨を返信

実装としては、Python3 を用い、Falcon と呼ばれる WebAPI の作成に特化したライブラリを使用した。機器 ID は、トークンと乱数を、sha256 と呼ばれるハッシュアルゴリズムを使用してハッシュ化した物を使用す

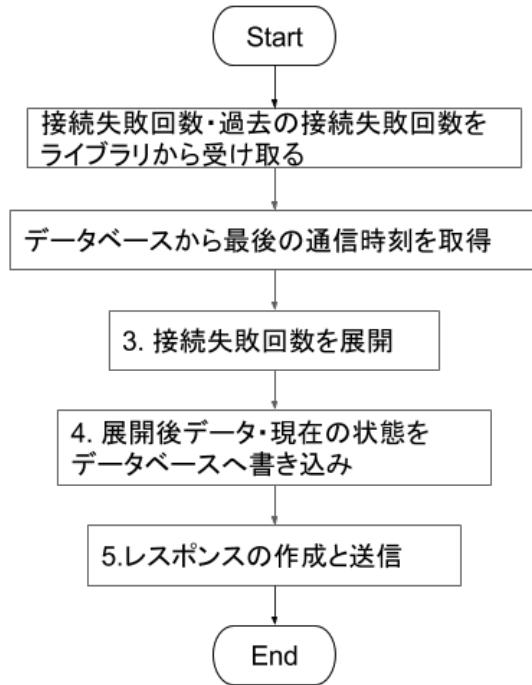


図 5.4: エージェントプログラム用インターフェースの動作（機器からの状態の通知があった際）

る。可視化アプリケーションへの登録は、機器情報データベースへの書き込みをもって、登録とすることとした。このプログラムを用いて、エージェントプログラムと通信を行うために、次のような URL を定義した。

- GET /deviceid
- POST /deviceid

先頭の GET・POST は、HTTP リクエストを表す。また、`/deviceid` は、各機器の機器 ID を表す。

5.5 エージェントプログラムとエージェントプログラム用インターフェース間の通信

エージェントプログラムとエージェントプログラム用インターフェース間の通信には、HTTPS を用いる。機器 ID の取得/発行・登録時は図 5.5 のような動作をし、機器状態の通知/監視の際は、図 5.6 のような動作をする。

各通信の書式は JavaScript Object Notation(JSON) という形式を用いる。トークンを「xxxxxxxx」、機器 ID を「yyyyyyyy」、過去にサーバへ接続できなかった回数と現在サーバへ接続できなかった回数をそれぞれ「pppp」・「qqqq」とすると、図中の各通信は次のようなメッセージとなる。

- 機器 ID の取得/発行・登録時の通信

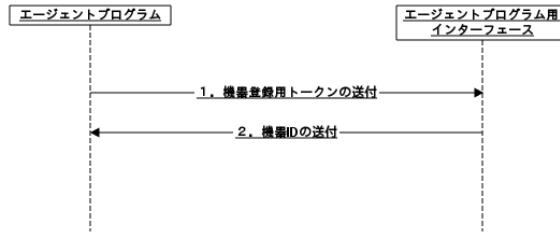


図 5.5: 機器 ID の取得/発行・登録時の通信

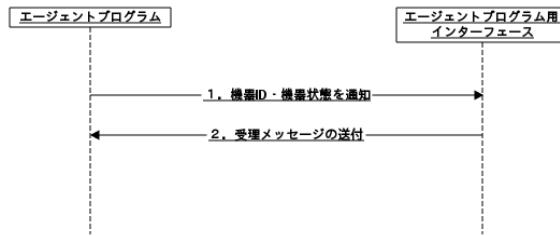


図 5.6: 機器状態の通知/監視時の通信

1. GET /deviceid

```
{ "token": "xxxxxxxxxx" }
```

2. { "deviceid": "yyyyyyyy" }

- 機器状態の通知/監視時の通信

1. POST /yyyyyyyy

```
{ "past": "pppp", "now": "qqqq" }
```

2. { "stat": "OK" }

また、トークンが不正であった場合や、デバイス ID が存在しない場合等は、サーバはデバイスに対して、HTTP Not Found(404) を送信する。

5.6 機器状態データベースの設計

機器状態データベースは、機器ごとに機器 ID と同一の名前のテーブルを作成し、機器状態を管理する。この機器 ID と同一の名前を持つテーブルを機器状態管理テーブルとする。この機器状態管理テーブルは、表 5.1 のような構造を持つ。

データベースは Influxdb を利用した。InfluxDB では、時刻を主キー (index) として扱うので、このような構造になっている。状態には、OK・NC が入り、OK は機器に異常が無かったこと、NC は、機器に異常は無

表 5.1: 機器状態管理テーブル

論理フィールド名	物理フィールド名	型
時刻	time	time
状態	stat	string

かったがサーバへ到達できなかった事を表す。また、各テーブルはエージェントプログラム用インターフェースが自動で作成し、可視化アプリケーションから、必要に応じて参照、削除を行う。

5.7 機器情報データベースの設計

機器情報データベースは、ユーザと機器の関係を格納するために使用される。ユーザテーブル、機器情報テーブルの2つのテーブルが有り、ユーザに関する情報と、機器に関する情報を分けて記録する。図 5.7 は、ユーザテーブルと機器情報テーブルの関連を表す。ユーザ 1 つに対し機器は 0 以上あり、機器 1 つに対しユーザは

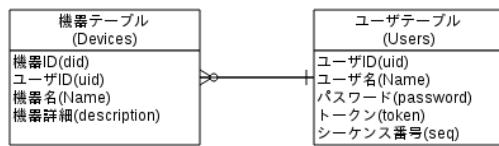


図 5.7: ユーザテーブルと機器情報テーブルの関連

1 つである。ユーザテーブルと機器テーブルは、ユーザ ID で紐付いている。

表 5.2 はユーザテーブルの構造を表している。ユーザテーブルでは、サービスへのログイン時に使用するユー

表 5.2: ユーザテーブル (Users)

論理フィールド名	物理フィールド名	型	制約
ユーザ ID	uid	integer	primary key
ユーザ名	Name	string	unique not null
パスワード	password	string	not null
トークン	token	string	unique
シーケンス番号	seq	integer	

ザ名とパスワードの他、内部で識別のため使用するユーザ ID、機器登録の際に使用されるトークン、機器 ID 生成の為に使用されるシーケンス番号が記録される。

表 5.3 は機器情報テーブルの構造を表している。

ユーザテーブル・機器テーブルは、エージェントプログラム用インターフェース・可視化アプリケーションから参照される。データベースには、sqlite3 を用いた。

表 5.3: 機器情報テーブル (Devices)

論理フィールド名	物理フィールド名	型	制約
ユーザ ID	uid	integer	foreign key primary key
機器 ID	did	string	primary key
機器名	name	string	not null
機器詳細	description	string	

5.8 可視化アプリケーションの設計

可視化アプリケーション次のような構造になっている。サーバサイドプログラムは Python3 を使用し, Flask という Web アプリケーションフレームワークを用いて作成した。クライアントサイドプログラムは, HTML/CSS , Javascript を使用し, Bootstrap,Jquery というライブラリを用いた。

5.9 サービス利用のイメージ

ユーザが本サービスを利用して行うことのできる動作は、つぎのようになっている。

5.9.1 機器の追加

機器の追加は、次のような操作にて、行うことができる。

1. サービスにログインする
2. サービスの画面からトークンの発行を押し、トークンを保存する。
3. 各機器に対して、エージェントプログラムのインストールと、トークンの設定を行う。
4. 機器の電源を入れる
5. 機器が追加された事を確認し、機器 ID を機器に対してラベル等を貼り付ける。
6. 順次他の機器に対して同様の操作を行う。

5.9.2 機器の削除

機器の削除の利用イメージは、次のようになっている。

1. サービスにログインする
2. サービスの画面から、該当の機器を削除する
3. 物理的な機器の撤去を行う

5.10 実装

時間的制約から，機器からの通知による監視のみを実装した．付録としてソースコードを添付する．ソースコードは [github](#) にも上がっている．

第6章 機器監視サービスの動作テスト

6.1 検証目的

IoT 機器の監視・管理の際の問題が解決できたのか、検証を行う。

- 各 IoT 機器の状態を監視するために、多数の IoT 機器へ設定をしなければいけない負担が軽減された事
- 増減や交換の度に、機器監視システムへの登録をしなければならない負担がなくなったこと
- 機器からの通知に基づいた監視ができている事

実装の都合から、この 3 つの要件の中で機器からの通知に基づいた監視のみの検証を行う。機器から、機器 ID と接続出来なかった回数等を、デバイス ID に紐付いた URL に送信し、監視サービスにて、監視することができる事を確認する。そのため、各 IoT 機器に対する設定・サーバでの登録は、手動で行う。

6.2 検証方法

検証は次のような小規模な IoT サービスを想定し、行った。IoT 機器の数は、1 台とし、RaspberryPi2 を使用することとした。OS は Raspbian jessie がインストールされていることとした。図 6.1 は使用した RaspberryPi2 と、IntelEdison である。今回は、IntelEdison は使用していないが、参考の為に上げた。

RaspberryPi2 は無線 LAN インターフェースを持たないので、パファロー製の無線 LAN インターフェースを使用した。期間は 2017 年 1 月 28 日正午から 1 時間行い、途中何度か IoT 機器の電源を抜き、正常に検知することを確認する。また、期間初めに、想定する IoT 機器の設定を行い、監視サービスにて発行した ID を設定する。

6.3 検証結果

本サービスを利用した監視について、作業別に確認した。

1. 設定

まず、サービス側で機器 ID の登録を行った。ブラウザからサービスへログインし、登録ボタンをクリックして、登録用ダイアログを開いた。ここで、登録用ダイアログに表示されている ID をコピーしておく。登



図 6.1: RaspberryPi2 と IntelEdison

録用ダイアログにて，機器名と機器の詳細（機器名を「RaspberryPi A」，機器の詳細を「TestDeviceA」とした）を入力し，登録ボタンを押した。その後，サービスの画面上に「RaspberryPi A」という名前を持つデバイスが作成されたことを確認した。図??は登録用ダイアログのスクリーンショットで，??は監視サービスメイン画面のスクリーンショットである。登録

次に，エージェントプログラムを機器の SD カードのホームディレクトリとなるディレクトリ（/home/pi/）へ，agent.sh という名前で書き込んだ。また，起動時にエージェントプログラムを実行するための設定ファイルを機器の SD カードの設定用ディレクトリ（/etc/systemd/system/）へ，devmon.service という名前で書き込んだ。この設定ファイルは，起動時に/home/pi/agent.sh のスクリプトを機器 ID を引数として実行するように指定している。

2. 運用

その後，機器の電源を入れ，機器上でエージェントプログラムが稼働していることを，systemctl status コマンドを用いて確認した。また，サービス側で指定した ID を持つ機器の状態が異常状態（稼働していないかネットワークから切断されている事を表す）から，正常状態（稼働していて，ネットワークが切断されていない事を表す）へ変化した事を確認した。機器から該当の ID に紐付いた URL に対し，通信が出来ている場合のみ，正常状態となるので，正常に ID と URL が紐付いている事が確認できたといえる。

3. トラブル検知

また，電源ケーブルを抜き，サービス上で該当の機器の状態が正常状態から異常状態へ変化することを確認した。変化には 2 分程度かかり，これは，機器が 1 分おきに通知を行っていることと，ブラウザ側のプログラムにて 1 分おきにサーバへ該当機器の状態を問い合わせているためである。再度電源ケーブ

ルを挿し直すと，1分程度でサービス上の機器の表示が異常状態から正常状態へ変化した．

これらから，監視対象機器と監視サーバが正常に連携し，機器の監視が行えることが分かった．

6.4 考察

提案する方法で IoT 機器からの監視が行えることは分かったが，下記問題については実装の都合上，確認が取れていらない．

- 各 IoT 機器の状態を監視するために，多数の IoT 機器へ設定をしなければいけない負担
- 増減や交換の度に，機器監視システムへの登録をしなければならない負担

今後，実装を進め，検証を行うこととする．

また，検証に置いて次のような改善点が見つかった．

- 過去の記録の表示

現在，過去の機器状態の記録については，文字記録として表示しているが，グラフ表示等の方が見やすいと感じた．また，期間を指定して閲覧できる機能も必要であることが分かった．

- アラート機能

監視サービスの画面を見続けるわけには行かないので，メール等によるアラート機能等を追加する必要があると感じた

- グループ表示機能

グループ表示等が行えると尚見やすい事が分かった．

- 一覧表示

機器が少量である場合は問題ないが，数千台となった場合に，一覧での表示は，見難い事が分かった．全体やグループでの稼働率等を表示し，確認が必要な機器のみを一覧で表示する等の工夫が必要となることがわかった．

また，本来ならば，IoT サービスを提供している企業に使用して頂き，評価を得る必要があったが，双方のスケジュールの都合と開発の遅れから行うことが出来なかった．しかし，今後 IoT サービスの開発が盛んになることや，使用する IoT 機器の数が多くなることから，本サービスの必要性は高くなっていくと考えられる．

第7章 考察

検証では、機器 ID と URL の組をサーバ側で管理することで、機器からの通知に基づいて監視可能であることを確認した。検証では、サーバへの機器の登録、機器への ID の付与は手動で行ったが、機器からの通知による監視、サーバ上で URL と ID の組み合わせの管理について、確認を行った。これにより、機器の追加・移動の際に、機器への設定の手間や、機器が接続されるネットワークの調査等の手間を省くことができた。また、機器への ID 付与や機器の登録の作業が負担となつたので、実装を進め、省力化を図り検証を行う事を考えている。

- サーバから機器 ID を付与することで、機器への設定を簡略化し、負担を軽減する事
- IoT 機器の変更や追加の際に、サーバに対し登録を行わなければならない負担が軽減された事

今後の課題として、次があげられる。

- IoT 機器のセキュリティ

現状、IoT 機器から監視サーバまでの間は、SSL を用いることで暗号化している。しかし、IoT 機器自身を分解することで、機器 ID を取得し、機器のなりすましが行われる危険がある。そのため、なんらかの方法で機器へのなりすましを検知し通知することや、機器に耐タンパ性を持たせる必要があるように考える。

- 物理との紐付けの改良

現状では、機器に機器 ID を貼り付ける事で解決しているが、現地に行った際にどの機器 ID が交換の必要があるのか、見比べなくてはならない。数十桁の機器 ID を一瞬で判別することは困難であり、改良の余地があるようだ。具体的には、QR コードを用いて、スマートフォン等から監視サービスの特定の機器の状態閲覧画面にアクセスすることができるようとする等が必要である。その際には、セキュリティの観点から、一般の利用者へ対する説明等の表示と、機器の管理者に対する表示を分ける必要が出てくると考えている。

また、現状では、一旦サービスに登録されたことを確認してからラベリングとしているが、QR コードを使用した場合、機器に対して予めラベリングを施しておくことでより設定の手間が省けるように感じている。具体的な実装としては、監視サービス側にて QR コードをまとめて発行し、ユーザは、QR コードを電源を入れる順番に貼り付けていく。機器側は、アクセスのあった順に発行した QR コードに埋め

込まれた機器 ID を割り振っていく。これにより、物理的なヒモ付の手間や、交換の際の手間等がより省けると感じている。

- 通知間隔の問題

機器から監視サーバへの通知は、現状では 1 分おきに行っているが、各種 IoT 機器にてこの 1 分が妥当なものなのか検証する必要があるように感じている。何故ならば、省電力を目的として、機器自体のスリープや、ネットワークからの離脱等を行う場合があるためである。このような場合、監視サーバにて通知間隔の設定を行うことで、誤認等を防ぐ必要がある。また、該当の機器の状態が不安定である場合など、意図的に監視間隔を短く設定することもできるよう考慮する必要があると考える。

- 機器の操作

RaspberryPi 等、IoT 機器によっては、電源が急に抜けた場合、記録していたデータが失われてしまう物がある。そのため、撤去の際に電源を切る操作を行ってから、電源を切る必要がある。しかし、多くの IoT 機器は、画面やキーボード等はついておらず、現地に画面やキーボードを運び接続するか、遠隔から操作する他無い。また、現地にエンジニアを派遣する必要もある。この手間を解決するために、機器の監視サービスの機能として、電源の OFF・再起動ボタンや、遠隔からのログイン等を行えるようにする必要も感じる。

- IoT サービスとの連携

IoT サービスは、収集したデータの分析や可視化を行う。そのため、IoT 機器が現在稼働しているのか、過去に置いて稼働していたのか、等を知る必要がある。現状では、分析などの際に手動で入力することや、ある程度の誤差を容認することで解決しているが、機器監視サービスが、監視情報を IoT サービスに対して提供することでも解決できるのではないかと考えている。そのために、監視サービスが IoT サービスとのインターフェースを持つ必要があると考えている。

第8章 おわりに

IoT とは、様々なモノにコンピュータを取り付け、インターネットを介して相互に情報をやり取りすることで、様々な自動化を図ろうという概念である。IoT サービスとは、IoT による利便性をユーザに提供するもので、IoT 機器とサーバのプログラムがインターネットを介して通信し合うことで成り立っている。IoT サービスの円滑な提供のためには、この構造を維持しなければならない。そのため、IoT 機器の監視が不可欠である。

本論文では、その IoT 機器の監視がサービス提供者にとって負担となっている事を取り上げた。株式会社ルナネクサスが提供している太陽光発電の発電量の監視のための IoT サービスを主にとりあげ、聞き取りを行った。その中で IoT 機器の監視が負担である事を問題として捉え、分析した。また、岡本商店街にて行った実験から、監視における技術的課題を明確にした。

その中でも、技術的制約から IoT 機器へ個別の ID を設定することの負担、監視サーバに対して監視対象機器を指定する負担を取り上げ、これら負担は、IoT 機器と監視サーバに対して、個別に整合性の取れた設定をしなければならない事が原因と考えた。

そこで、本研究では、監視サーバにて、設定を一元的に管理することで、これら問題を解決した。各機器が接続されるネットワークが多様である問題を、IoT 機器から通知を送ることで解決し、各機器への設定と監視サーバへの登録が負担である問題を、監視サーバにて各 IoT 機器の設定を管理することで解決した。

この機器監視サービスを実現する為、IoT 機器にインストールされるエージェントプログラム、機器監視サーバ上で動作するエージェントプログラム用インターフェース、Web アプリケーションを作成した。

また、要件を満たしているか検証を行い、IoT 機器の動作状態について監視できていることを確認した。今後の課題として、IoT 機器への設定の簡略化や、ユーザインターフェースの向上、様々な IoT 機器への対応があることがわかった。本来ならば株式会社ルナネクサスにて使用してもらい評価を得るべきところではあるが、スケジュールの都合により、実現しなかった。しかし、今後 IoT サービスの普及がより進み、使用する IoT 機器の数が多くなることから、本サービスが求められることが推測される。

謝辞

本論文は、著者が神戸情報大学院大学情報技術研究科情報技術専攻在学中に、横山研究室にて行った研究をまとめたものです。本研究に関してご指導ご鞭撻を頂きました横山輝明講師に心より感謝申し上げます。また、本論文をご精読頂き、有用なコメントと励ましをくださった藤原明生准教授に深謝致します。

本研究を進めるにあたって、インタビューや訪問を受け入れてくださった、株式会社ルナネクサス 藤戸様に感謝申し上げます。そして、論文を書く際にアドバイスをくださった、嶋教授と、嶋研究室の渡邊香織さん、田頭潤さん、笠谷拓伸さん、また、研究についてアドバイスを頂いた横山研究室卒業生の鄒曉明さんと、良き議論相手である京都産業大学大学院修士1年目の石原真太郎さんに感謝致します。

参考文献

- [1] トレンド・イノベーション 稲田修一 「ビッグデータ活用でビジネスはどう変わったか～コマツにおけるモノのインターネット事例から考える～」 <https://www.salesforce.com/jp/blog/2013/12/vol3-bigdata.html> (2017年2月10日 閲覧)
- [2] Richo JAPAN Corp. 「出力機器のリモート管理サービス「@Remote」」 <https://www.ricoh.co.jp/remote/> (2017年2月10日 閲覧)
- [3] 日経テクノロジーオンライン 高野 敦 「[生体センシング] "着る "センサーで健康情報を計測」 <http://itpro.nikkeibp.co.jp/article/COLUMN/20140526/559230/> (2017年2月10日 閲覧)
- [4] 株式会社 SORACOM 「SORACOM の概要」 <https://soracom.jp/overview/> (2017年2月7日 閲覧)
- [5] TechCrunch Japan 「【詳報】ソラコムがペールを脱いだ、月額300円からのIoT向けMVNOサービスの狙いとは？」 <http://jp.techcrunch.com/2015/09/30/soracom-launches-mvno-service-for-iot/> (2017年2月7日 閲覧)
- [6] 株式会社沖縄アイオー 金城辰一郎 「ソラコムによるIoTサービス内容とは？非エンジニアがその革新的な魅力と導入事例をわかりやすく徹底解説」 http://okinawa.io/blog/tech/soracom_iot (2017年2月7日 閲覧)
- [7] IoTNews.jp 小泉耕二 「【前編】SORACOMが発表した新サービスは、なにがすごいのか？ SORACOM Connected」 <https://iotnews.jp/archives/12037> (2017年2月7日 閲覧)
- [8] THE BRIDGE Takeshi Hirano 「SORACOMの凄さは第三者が「SIM」を自由に発行・運用できるこどーーIoT向けモバイル通信PF、ソラコムが提供開始」 <http://thebridge.jp/2015/09/soracom> (2017年2月7日 閲覧)

添付資料 岡本商店街での事例

実験概要

2015年12月8日から2016年2月26日まで、NPO法人コミュニティリンクへのインターンシップの一環として、岡本商店街にて人流観測を行った。岡本商店街とは、神戸市東灘区にある阪急岡本駅とJR摂津本山駅の間に位置する商店街のことである。実験は、商店街の方に人流を可視化するIoTサービスを提供し、商店街の活性化に役立てるといった趣旨で行った。観測は、2016年2月7日から2016年3月14日まで行った。

人流観測とは、各地点から各地点迄をある時に移動した人数を観測するものである。通常は、観察員がカウントを用いて数えるが、それでは各地点間を移動した人数はわかるが、その人が以前どの地点に居たのかはわからない。そこで、携帯電話についているWifi機能を利用し、観測を行うこととした。携帯電話のWifi機能は、無線LANの接続に使われるが、接続毎にWifi機能を有効にすることが手間なため、常時ONにしている人も少なくない。携帯電話のWifi機能を有効にしている場合、携帯電話から接続可能な無線LANを探す為、プローブパケットというものが定期的に送出される。プローブパケットには、そのプローブパケットを送出した機器の物理アドレスが含まれており、個々の機器が識別可能である。そのプローブパケットを複数地点で観測し、含まれている物理アドレスと受信時刻を照合することで、携帯電話端末を持った人がどのように移動をしたのかが分かる。岡本商店街では、この原理を利用して、人流観測を行った。

観測・分析・可視化システムの構成

岡本商店街の5店舗に開発した観測機器を設置し、サーバにて蓄積・分析・可視化を行った。構成としては、観測機器内で動作するプログラムが定期的にサーバに観測データを送信する。サーバ上では、分析プログラムと可視化プログラムが動作しており、観測データは分析プログラムへ渡される。分析プログラムは、観測データを可視化プログラムの要求によって分析し、結果を可視化プログラムへ渡す。可視化プログラムは、ユーザからの操作によって分析プログラムを呼び出し、分析結果をWebインターフェースによって可視化する。このシステムのユーザは、ブラウザからWebインターフェースへアクセスすることで、操作・分析結果の閲覧をすることができる。図8.1は、開発したシステムの構成を示している。

開発した観測機器は、RaspberryPiとBaffalo製のWifiドングル、ampsence・rsyncというソフトウェアを利用し作成した。図8.2は、開発した観測機器である。RaspberryPiとは、小型PCの一つで安価入手が可能である。また、利用したRaspberryPiにはWifiインターフェースが存在しなかったので、Baffalo製のUSB

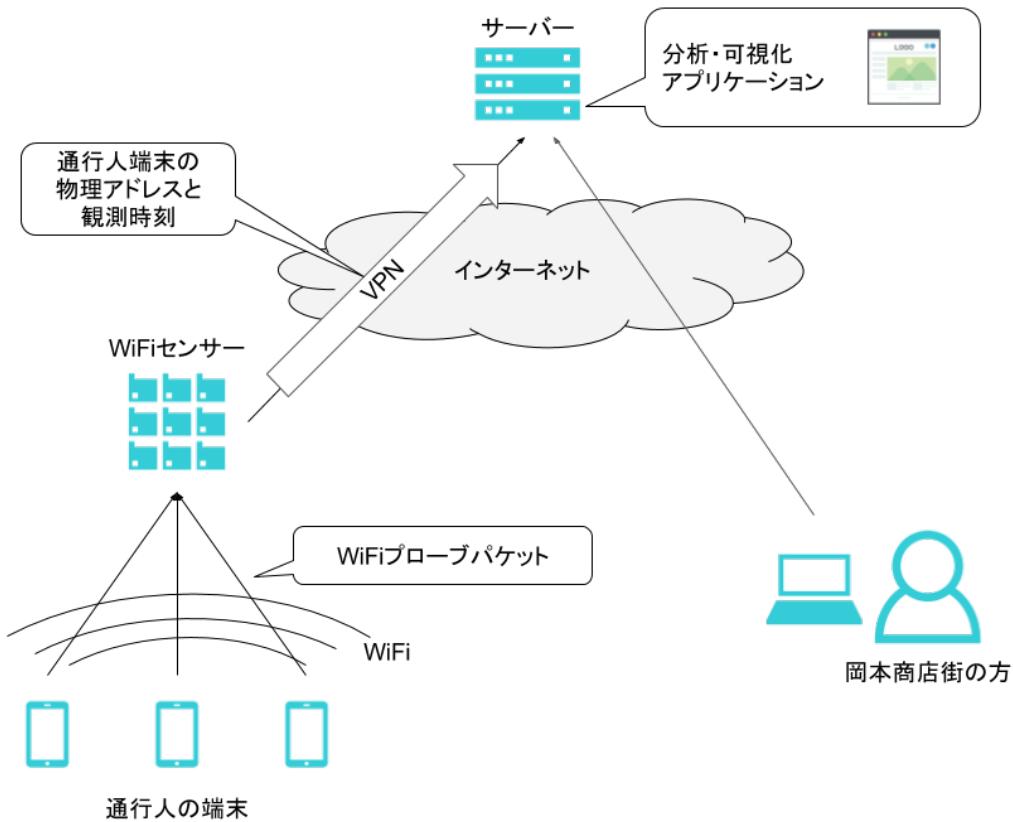


図 8.1: 岡本商店街人流観測 構成図

接続が可能な WiFi ドングルを接続した。ampsence とは、プローブパケットを受信しプローブパケットに含まれる物理アドレスと観測日時をファイルへ記録するソフトウェアである。rsync とは、ディレクトリの中を同期させるプログラムで、遠隔にあるコンピュータの指定したディレクトリの中と、手元のコンピュータの指定したディレクトリを同期することができる。開発した観測機器は、起動時に ampsence を動作させ、定期的に rsync を実行するよう設定した。これによって、各所に設置された観測機器の情報をサーバ上に集約・蓄積する。

作成した分析プログラムは、可視化プログラムから指定された期間内の観測データを読み込み、地点ごと観測時間ごとの物理アドレスの数の推移・地点ごと観測時間ごとの移動した人数の割合を集計し、可視化プログラムへ渡す。作成した可視化プログラムは、ユーザから指定された期間を分析プログラムへ渡し、分析結果を棒グラフや円グラフといった形で、ユーザへ表示する。図 8.3 は作成した Web アプリケーションである。

観測機器からサーバまでのネットワークは、観測した物理アドレスが流れるという点、ソフトウェアのアップデートの為にログイン可能にしたいといった要望から、VPN を使用した。VPN には、OpenVPN を使用した。そのため、観測機器とサーバに OpenVPNClient と OpenVPNServer をそれぞれインストールし、設定を行った。

また、観測機器、分析・可視化アプリケーションを開発するにあたって、次のような点を考慮した。

- ネットワーク機器の設定が不要であること



図 8.2: 岡本商店街人流観測 使用した機器

機器が設置されるネットワークは、店舗のネットワークである。そのため、ネットワーク機器の設定を変更することはできない。よって、観測機器と分析・可視化アプリケーションの連携が、ネットワーク機器の設定に左右されないよう設計する必要があった。

- トラブル対応のしやすさ

機器が設置される場所は、岡本商店街内の店舗である。そのため、設置やトラブル対応、回収については、予め店舗に連絡を入れなくてはならない。また、夜間や店舗の休日に対応することは困難である為、遠隔から観測機器にログインする必要があった。

- 設置箇所の問題

店舗によっては、店舗内に無線ネットワークが存在しなかったので、後述する SORACOM Air を用いることとした。

- データ損失に備える

サーバ側の不良による観測データの損失に備え、機器自体にも情報を蓄積することとした。

考察

実験では、次のようなトラブルがあった。

- 設置後、電源が抜けており、観測ができていないことがあった
- 設定ミスにより、観測できていなかったことがあった

岡本商店街動態情報可視化システム

2016年 1月 1日 現在の動態情報



図 8.3: 岡本商店街人流観測可視化アプリケーションスクリーンショット

その結果、次のような問題が起きた。

- 分析時に、観測できていない期間を推測する必要があった
- 観測データの不足によって、曜日ごとの来客数の動向等の分析を行うことが出来なかった
- 何度も現地に行き、確認を行わなければならなかった

その為、観測機器の監視と、観測機器ごとの設定の簡略化が必要であると考えた。しかし、観測機器の監視には次のような技術的困難や制約があり、行うことができなかった。

- 機器が接続するネットワークの設定を変更することは出来ない
- 店舗ネットワーク及びSORACOMAir のネットワークでは、プライベートアドレスが利用されているため、インターネットから観測機器にアクセスすることが出来ない
- 提供しているサービスに監視機能を組み込むのは難しい
- 新規に監視サーバを立ち上げるのは負担となる

添付資料 ソースコード

ソースコード概要

作成したソースコードはgithubにもアップロードされている .<https://github.com/miyasakakoki/devicemonit/tree/feature>

作成したプログラムは大きく分けて 3 つある . プログラム名の前に付いているのは , ファイル名である .

- エージェントプログラム

agent/agent.sh エージェントプログラム

- エージェントプログラム用インターフェース

api/api.py エージェントプログラム用インターフェース本体

- 監視アプリケーション

ui/devicemonit.py 監視アプリケーション本体

ui/schema.sql 機器情報データベース作成用スクリプト

ui/templates/outerlayout.html Web ページ全体のレイアウト用テンプレート

ui/templates/layout.html ログインページを除く Web ページ用のテンプレート

ui/templates/login.html ログインページ

ui/templates/dashboard.html 機器情報一覧ページ

ui/templates/devicelog.html 過去の機器状態一覧ページ

エージェントプログラム (agent/agent.sh)

このプログラムは , IoT 機器にインストールされる . シェルスクリプトとして書かれている実行の際 , 引数として , 機器 ID を指定する .

ソースコード 8.1: エージェントプログラム

```
1 #!/bin/bash
2 URL="https://210.146.64.147/UserName/kariID"
3
4 if [ ! -e ./DeviceID ]; then
5     #Get DeviceID
6     curl -k ${URL} >> ./DeviceID
7 fi
8
9 #Read DeviceID
10 DeviceID='cat ./DeviceID'
11
12 seq=0
13 if cat log 2>/dev/null 1>/dev/null; then
14     log='cat log'
15 else
16     log=0
17 fi
18 while true; do
19     dat='echo "{\"seq\":$seq,\"stat\":\"OK\",\"log\":{\"seq\":$log}}"'
20     echo $dat
21     tmp='curl -X POST -d "${dat}" http://${1}/${2} -sS -m 1 -H "Accept:application/json"
-H "Content-type:application/json"'
22     if [ $? -eq "0" ]; then
23         seq=0
24         rm log 2> /dev/null
25     else
26         seq=$((++seq))
27         if [ ${log} -eq 0 ]; then
28             echo ${seq}>log
29         fi
30     fi
31     cmd='echo ${tmp} | sed "s/^.*'command':\"\\\'\([^\']*\')\".*$/\\1/g"'
32     echo ${cmd}
33     if [ "${cmd}" = "shutdown" ]; then
34         echo shutdown -h now
35     fi
36     if [ "${cmd}" = "reboot" ]; then
37         echo reboot
38     fi
39     sleep 60
40 done
```

エージェントプログラム用インターフェース (api/api.py)

このプログラムは、サーバ上で動作する。プログラムを作動させるには、Python3,gunicorn 等の WSGI サーバがインストールされている必要がある。また、ライブラリとして Falcon,InfluxDBClient がインストールされている必要がある。gunicorn を用いた際の起動の仕方は次のようにになっている。

以下にソースコードをあげる。

ソースコード 8.2: エージェントプログラム用インターフェース

```
1#!/usr/bin/python3
2
3 import falcon
4 from influxdb import InfluxDBClient
5 import time
6 import json
7
8 host='localhost'
9 port=8086
10 user='test'
11 password='mytestuser'
```

```

12 dbname='devicemonit'
13
14 class MyAPI( object ):
15     def on_post( self, req, res, id ):
16         if len(id) < 8:
17             raise falcon.HTTPNotFound()
18         buf = req.stream.read()
19         data = json.loads( buf.decode('utf-8') )
20         if not( "seq" in data and "stat" in data ):
21             raise falcon.HTTPNotAccepted()
22         cli = InfluxDBClient( host, port, user, password, dbname )
23         ret = cli.query( "show\u005fmeasurements\u005fwith\u005fmeasurement\u003d\"{0}\\";.format( id ) )
24         tmp = []
25         debugp = ""
26         if len( ret.raw ) < 1:
27             raise falcon.HTTPNotFound()
28         if "log" in data:
29             if "seq" in data["log"] and data["log"]["seq"] != 0:
30                 ret = list(cli.query( "select\u005flast(Stat)\u005ffrom\u005f\"{0}\\";\u005fwhere\u005ftype\u003d\"s\"\u005f" ".format( id ), epoch="s" ))[0] #Get last timestamp
31             if len( ret ) > 0:
32                 lasttime = int(ret[0]["time"])
33                 for i in range(data["log"]["seq"]):
34                     lasttime += 60
35                     tmp.append( {"measurement":id, "tags":{}, "time":lasttime*1000000000, "fields":{"Stat":"NC"} } )
36             now = int(time.time())
37             lasttime = now-60
38             for i in range( data["seq"] ):
39                 tmp.append( {"measurement":id, "tags":{}, "time":lasttime*1000000000, "fields": {"Stat":"NC"} } )
40             lasttime -= 60
41             tmp.append( {"measurement":id, "tags":{}, "time":now*1000000000, "fields": {"Stat":"OK"} } )
42             #check command
43             ret=list(cli.query( "select\u005flast(Stat)\u005ffrom\u005f\"{0}\\";.format( id ) ))[0][0]
44             if ret["last"] == "OK" or ret["last"] == "NC":
45                 com = ""
46             else:
47                 com = ret["last"]
48             cli.write_points( tmp )
49             res.status = falcon.HTTP_200
50             res.body = str( {"stat":"OK", "time":now, "command":com, "debug":debugp } )
51             res.content_type= "application/json"
52 app = falcon.API()
53 app.add_route( '/{id}', MyAPI() )

```

監視アプリケーション(ui/)

監視アプリケーションは，Flask を用いたサーバサイドプログラムと，Web ブラウザ上で動作する Javascript が埋め込まれた Web ページに分けることができる。Web ページは，Flask に内蔵されているレンダリングエンジンによって，複数のテンプレートファイルから生成される。アプリケーションの起動には，Python3 がインストールされている必要がある。また，Flask,InfluxDBClient,sqlite3 がインストールされている必要がある。サーバマシン上で次の操作を行うことで起動することができる。

監視アプリケーション本体 (ui/devicemonit.py)

このプログラムは、サーバサイドで動作するプログラムである。データベースの初期化の為に機器情報データベース作成用スクリプトを読み込み、実行している。

ソースコード 8.3: 監視アプリケーション本体

```
1 #!/usr/bin/python3
2
3 from flask import Flask, request, g, send_from_directory, render_template, session, redirect, url_for, jsonify
4 from functools import wraps
5 import sqlite3
6 from influxdb import InfluxDBClient
7 import string
8 import os
9 import datetime
10 import random
11
12 app = Flask(__name__)
13 app.config.from_object(__name__)
14 app.config.update(dict(
15     DATABASE = os.path.join(app.root_path, 'devmon.db'),
16     SECRET_KEY = 'devmonit',
17     USERNAME = 'admin',
18     PASSWORD = 'default',
19     INFLUXDB = {
20         "HOST": "localhost",
21         "PORT": 8086,
22         "USER": "test",
23         "PASS": "mytestuser",
24         "NAME": "devicemonit"
25     }
26 ))
27 app.config.from_envvar('FLASKR_SETTINGS', silent=True)
28 #app.config['SECRET_KEY'] = os.urandom(24)
29 app.config['SECRET_KEY'] = "testkey"
30
31 def condb():
32     """Connect to Database"""
33     rv = sqlite3.connect(app.config['DATABASE'])
34     rv.row_factory = sqlite3.Row
35     return rv
36 def getdb():
37     if not hasattr(g, 'sqlite_db'):
38         g.sqlite_db = condb()
39     return g.sqlite_db
40
41 def initdb():
42     db = getdb()
43     with app.open_resource('schema.sql', mode='r') as f:
44         db.cursor().executescript(f.read())
45     db.commit()
46
47 def checklogin( email, password ):
48     cur = getdb().cursor()
49     cur.execute("select * from User where email=? and pass=?;", (email,password) )
50     ret = cur.fetchone()
51     if ret is None:
52         return {"stat":"None"}
53     return {"stat":"OK", "uid":ret["uid"], "Name":ret["Name"] }
54
55 def get_all( uid ):
56     ifdb = app.config["INFLUXDB"]
57     client = InfluxDBClient( ifdb["HOST"], ifdb["PORT"], ifdb["USER"], ifdb["PASS"], ifdb["NAME"] )
58     cur = getdb().cursor()
59     cur.execute("select did, Name, Description from Devices where uid=?;", (uid,) )
60     result = cur.fetchall()
61     ret = []
62     if result is not None:
63         for i in result:
64             tmp = { "ID": i["did"], "Name": i["Name"], "Description": i["Description"] }
```

```

65         rs = list(client.query( "select\u00a0last(Stat)\u00a0from\u00a0\"{0}\u00a0where\u00a0type\u00a0<>\u00a0",
66             command ;".format( i["did"] ), epoch="s" ).get_points())
67         now = int(datetime.datetime.now().timestamp())
68         if len(rs) == 0 or rs[0]["time"] == 0:
69             tmp["Stat"] = "NG"
70             tmp["time"] = "0"
71         else:
72             tmp["time"] = datetime.datetime.fromtimestamp( rs[0]["time"] ).strftime( "
73                 %Y/%m/%d\u00a0%H:%M:%S" )
74             tmp["Stat"] = "NG" if rs[0]["time"] < now-65 else "OK"
75             ret.append( tmp )
76     return ret
77
78 def checkdeviceid( did ):
79     if did == None or len(did) < 8:
80         return False
81     for c in did:
82         if c not in string.ascii_letters and c not in string.digits and c not in "_#":
83             return False
84     r = getdb().execute( "select\u00a0*\u00a0from\u00a0devices\u00a0where\u00a0did\u00a0=\u00a0?", (did,) ).fetchone()
85     return False if r is not None else True
86
87 def login_required(f):
88     @wraps(f)
89     def decorated_function( *args, **kwargs ):
90         if "uid" not in session:
91             return redirect( url_for("login") )
92         session["Date"] = datetime.datetime.now()
93         return f( *args, **kwargs )
94     return decorated_function
95
96 @app.cli.command('initdb')
97 def initdb_command():
98     """Initializes the database"""
99     initdb()
100    print( "The database initialized." )
101
102 @app.teardown_appcontext
103 def closedb(error):
104     if hasattr( g, 'sqlite_db' ):
105         g.sqlite_db.close()
106
107
108 @app.route( "/", methods=["GET"] )
109 def test():
110     return "OK"
111
112 @app.route( "/login", methods=["GET"] )
113 def login_page():
114     return render_template( 'login.html' )
115
116 @app.route( "/login", methods=["POST"] )
117 def login():
118     if "uid" in session:
119         return redirect( url_for( "dashboard_page" ) )
120     f = request.form
121     if "email" in f and len(f["email"]) > 2 and "password" in f and len(f["password"]) > 2:
122         res = checklogin( email=f["email"], password=f["password"] )
123         if res["stat"] == "OK":
124             session["uid"] = res["uid"]
125             session["Name"] = res["Name"]
126             session["Date"] = datetime.datetime.now()
127             return redirect( url_for("dashboard_page") )
128         error = "Invalid credentials. Please try again."
129     return render_template( 'login.html', error=error )
130
131 @app.route( "/logout" )
132 def logout():
133     session.pop( "uid", None )
134     return redirect( url_for( "login" ) )
135
136

```

```

137 @app.route( "/dashboard", methods=["GET"] )
138 @login_required
139 def dashboard_page():
140     return render_template( 'dashboard.html', page="summary" )
141
142 @app.route( "/devicelog", methods=["GET"] )
143 @login_required
144 def devicelog_page():
145     return render_template( 'devicelog.html', page="devicelog" )
146
147 @app.route( "/api/device/all" )
148 @login_required
149 def devicestatus_all():
150     return jsonify( { "devices": get_all( session["uid"] ), "Time": session["Date"].strftime( "%Y/%m/%d %H:%M:%S" ) } )
151
152 @app.route( "/api/deviceID" , methods=["GET"] )
153 @login_required
154 def gen_device_id():
155     while True:
156         tmp = "".join( random.SystemRandom().choice( string.ascii_letters + string.digits ) for _ in range(16) )
157         result = getdb().cursor().execute( "select * from devices where did=?;", (tmp,) ).fetchone()
158         if result == None:
159             break;
160     return jsonify( {"ID":tmp} )
161
162 @app.route( "/api/deviceID" , methods=["POST"] )
163 @login_required
164 def check_device_id():
165     return jsonify( { "stat": "OK" if checkdeviceid( request.json["ID"] ) else "NG" } )
166
167 @app.route( "/api/device/<DeviceID>" , methods=["POST"] )
168 @login_required
169 def mod_device( DeviceID ):
170     if request.json["stat"] == "new":
171         if checkdeviceid(DeviceID):
172             db = getdb()
173             db.execute( "insert into devices (uid,did,Name,Description) values (?,?,?,?);", (session["uid"], DeviceID, request.json["Name"], request.json["Description"] ) )
174             db.commit()
175             ifdb = app.config["INFLUXDB"]
176             client = InfluxDBClient( ifdb["HOST"], ifdb["PORT"], ifdb["USER"], ifdb["PASS"], ifdb["NAME"] )
177             client.write_points( [ { "measurement":DeviceID, "tags":{}, "time":0, "fields":{ "Stat":"None" } } ] )
178             return jsonify( {"stat":"OK"} )
179         return jsonify( {"stat":"NG"} )
180     else:
181         db = getdb()
182         db.execute( "update devices set Name=?, Description=? where did=?;", (request.json["Name"], request.json["Description"], DeviceID ) )
183         db.commit()
184         return jsonify( {"stat":"OK"} )
185
186 @app.route( "/api/device/<DeviceID>" , methods=["DELETE"] )
187 @login_required
188 def del_device( DeviceID ):
189     db = getdb()
190     db.execute( "delete from devices where did=?;", (DeviceID,) )
191     db.commit()
192     ifdb = app.config["INFLUXDB"]
193     client = InfluxDBClient( ifdb["HOST"], ifdb["PORT"], ifdb["USER"], ifdb["PASS"], ifdb["NAME"] )
194     client.query( "Drop measurement \\"{}\\";".format( DeviceID ) )
195     return jsonify( {"stat":"OK"} )
196
197 @app.route( "/api/device/<DeviceID>/power" , methods=["POST"] )
198 @login_required
199 def power( DeviceID ):
200     db = getdb()
201     ret = db.execute( "select * from Devices where did=?;", (DeviceID,) ).fetchone()
202     if len( ret ) < 1:

```

```

203     return jsonify( {"stat":"NG"} )
204     ifdb = app.config["INFLUXDB"]
205     cli = InfluxDBClient( ifdb["HOST"], ifdb["PORT"], ifdb["USER"], ifdb["PASS"], ifdb["NAME"] )
206     ret = list(cli.query( "select\u00a9last(Stat)\u00a9from\u00a9\"{0}\u00a9;\".format(DeviceID) ) )[0]
207     if ret[0]['last'] != "NC" and ret[0]['last'] != "OK":
208         return jsonify( {"stat":"NG"} )
209     else:
210         if request.json["command"] == "shutdown" or request.json["command"] == "reboot":
211             cli.write_points( [{"measurement":DeviceID,"tags":{"type":"command"},"time":int(
212                 datetime.datetime.now().timestamp())*1000000000, "fields":{"Stat":request.
213                 json["command"]}}] )
214     return jsonify( {"stat":"OK"} )
215
216 @app.route( "/api/device/<DeviceID>/log", methods=["GET"] )
217 @login_required
218 def devicelog( DeviceID ):
219     ifdb = app.config["INFLUXDB"]
220     cli = InfluxDBClient( ifdb["HOST"], ifdb["PORT"], ifdb["USER"], ifdb["PASS"], ifdb["NAME"] )
221     ret = list( cli.query( "show\u00a9measurements\u00a9with\u00a9measurement\u00a9=\u00a9\"{0}\u00a9;\".format( DeviceID ) ) )
222     if len( ret ) < 1:
223         return jsonify( { "data":[], "command":[], "stat": "NG" } )
224     ret = list( cli.query( "select\u00a9Stat\u00a9from\u00a9\"{0}\u00a9where\u00a9type\u00a9=>\u00a9'command'\u00a9;.format( DeviceID ),
225         epoch='s' ) )[0]
226     tmp = []
227     rec = None
228     for item in ret:
229         item['time'] = int(item['time']/60)*60
230     for item in ret:
231         if rec == None:
232             if item['time'] == 0:
233                 continue
234             rec = { 'stime': item['time'], 'etime': item['time'], 'stat': item['Stat'] }
235         if rec['etime'] == item['time']:
236             continue
237         if rec['etime']+60 == item['time']:
238             if rec['stat'] == item['Stat']:
239                 rec['etime'] = item['time']
240             else:
241                 tmp.append( rec )
242                 rec = { 'stime': item['time'], 'etime': item['time'], 'stat': item['Stat'] }
243         else:
244             tmp.append( rec )
245             tmp.append( { 'stime': rec['etime']+60, 'etime': item['time']-60, 'stat': "NG" } )
246             rec = { 'stime': item['time'], 'etime': item['time'], 'stat': item['Stat'] }
247     tmp.append( rec )
248     for item in tmp:
249         if item['stat'] == "OK":
250             item['stat'] = 2
251         elif item['stat'] == "NC":
252             item['stat'] = 1
253         else:
254             item['stat'] = 0
255     ret = list( cli.query( "select\u00a9Stat\u00a9from\u00a9\"{0}\u00a9where\u00a9type\u00a9=\u00a9'command'\u00a9;.format( DeviceID ),
256         epoch='s' ) )
257     ret = [] if len( ret ) == 0 else ret[0]
258     return jsonify( { 'data':tmp, 'command':tmp, 'ID':DeviceID} )
259
260 """
261 [
262     {'time': "yyyy/mm/dd hh/mm/ss", 'Stat':2},
263     {'time': "yyyy/mm/dd hh/mm/ss", 'Stat':1},
264     {'time': "yyyy/mm/dd hh/mm/ss", 'Stat':0},
265 ]
266 """
267
268 @app.route( "/signup", methods=["GET"] )
269 def signup_page():
270     return render_template( 'signup.html' )

```

```
271
272 @app.route( "/signup", methods=["POST"] )
273 def signup():
274     t = request.get_json(force=True)
275     return "OK"
276
277
278 @app.route('/c3/<path:path>')
279 def send_js(path):
280     return send_from_directory('statics', path)
281
282 if __name__ == "__main__":
283     app.run()
```

機器情報データベース作成用スクリプト (ui/schema.sql)

このスクリプトは、データベースの初期化の際に、アプリケーション本体によって読み込まれる。

ソースコード 8.4: 機器情報データベース作成用スクリプト

```
1 PRAGMA foreign_keys = ON;
2
3 drop table if exists User;
4 create table User(
5     uid integer primary key autoincrement,
6     email text not null,
7     name text not null,
8     pass text not null
9 );
10
11 insert into User( email, name, pass ) values( "asdf@asdf", "TestUser", "password" );
12
13 drop table if exists devices;
14 create table devices(
15     uid integer,
16     did text not null,
17     name text not null,
18     description text not null,
19     primary key( uid, did ),
20     foreign key( uid ) references User(uid) on delete cascade on update cascade
21 );
```

テンプレートファイル郡

これらは、利用者がブラウザからアプリケーションにアクセスされた際に、Flask のレンダリングエンジン (jinja) によって読み込まれ、処理される。Javascript、HTML/CSS と、jinja 用スクリプトによって書かれている。

ui/templates/outerlayout.html Web ページ全体のレイアウト用テンプレート

このテンプレートファイルは、アプリケーションが出力する全ての Web ページに適応される。

ソースコード 8.5: 全体のレイアウト用テンプレート

```
1 <!DOCTYPE html>
2 <html lang="en">
```

```

3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width,initial-scale=1">
7      <meta name="description" content="">
8      <meta name="author" content="">
9      <title>Devicemonit</title>
10     {% block style %}
11     {% endblock %}
12
13     <!-- Latest compiled and minified CSS -->
14     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/
15         bootstrap.min.css" integrity="sha384-
16             BVYiiSIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="
17             anonymous">
18             <!-- Optional theme -->
19             <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/
20                 bootstrap-theme.min.css" integrity="sha384-
21                     rHyO1iRsVXV4nD0JutlnGas1CJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXw1/Sp" crossorigin="
22                     anonymous">
23             <!-- C3js -->
24             <link rel="stylesheet" href="c3/c3.css">
25
26     </head>
27     <body>
28         {% block content %}
29         {% endblock %}
30
31     <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></
32         script>
33
34         <!-- Latest compiled and minified JavaScript -->
35         <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
36             integrity="sha384-
37                 Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnPJA712mCWNIPG9mGCD8wGNICPD7Txa" crossorigin="
38                 anonymous"></script>
39
40         <!-- for C3js -->
41         <script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
42         <script src="/c3/c3.min.js" charset="utf-8"></script>
43
44         {% block script %}
45         {% endblock %}
46
47     </body>
48
49 </html>

```

ui/templates/layout.html ログインページを除く Web ページ用のテンプレート

このテンプレートファイルは、ログインページ以外の Web ページに適応される。

ソースコード 8.6: レイアウト用テンプレート

```

1  {% extends "outerlayout.html" %}
2  {% block content %}
3  <nav class="navbar navbar-default navbar-fixed-top">
4      <div class="container">
5          <div class="navbar-header"><a class="navbar-brand" href="#">Test project</a>
6              </div>
7          <div id="navbar" class="navbar-collapse collapse">
8              <ul class="nav navbar-nav">
9                  <li {% if page == "summary" %} class="active" {% endif %}><a
10                      href="{{url_for('dashboard_page')}}">Summary</a></
11                      li>
12                  <li {% if page == "devicelog" %} class="active" {% endif %}><a
13                      href="{{url_for('devicelog_page')}}">Devices Log</
14                      a></li>
15                  <li {% if page == "alerts" %} class="active" {% endif %}><a
16                      href="#">Alert List</a></li>
17
18              </ul>
19              <ul class="nav navbar-nav navbar-right">
20                  <li><a href="/logout">Logout</a></li>
21
22          </div>
23
24 </div>

```

```
17 </nav>
18
19 {% block maincontent %}
20 {% endblock %}
21
22 {% endblock %}
```

ui/templates/login.html ログインページ

ソースコード 8.7: ログインページ

```
1  {% extends "outerlayout.html" %}
2  {% block content %}
3  <div class="container">
4      <form class="form-signin" action="/login" method="POST" id="signin">
5          <h2 class="form-signin-heading">Sign in...</h2>
6          <label for="inputEmail" class="sr-only">Email address</label>
7          <input type="email" name="email" id="inputEmail" class="form-control" placeholder="Email address" required="" autofocus="" />
8          <label for="inputPassword" class="sr-only">Password</label>
9          <input type="password" name="password" id="inputPassword" class="form-control" placeholder="Password" required="" />
10         <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
11         <div class="form-signin-footing">
12             <a href="/create"> Create new Project....</a>
13         </div>
14     </form>
15     {% if error %}
16         <div class="alert alert-danger">
17             <strong>Error:</strong> {{error}}
18         </div>
19     {% endif %}
20 </div>
21 {% endblock %}
22
23 {% block style %}
24 <style>
25 #signin {
26     max-width:400px;
27     margin: 16px auto;
28 }
29 .alert {
30     max-width:400px;
31     margin: 16px auto;
32 }
33 </style>
34 {% endblock %}
35
36 {% block script %}
37 <script>
38 //$( function (){
39 //    $("#signin").submit( function(event){
40 //        event.preventDefault();
41 //        var email = $( "#inputEmail" ).val();
42 //        var passw = $( "#inputPassword" ).val();
43 //        $.post( $(this).attr("action"), JSON.stringify({ Email:email, Pass:passw }), 
44 //            function( data ) {
45 //                if( data["Stat"] == "OK" ) $( location ).attr( "href", "/dashbord" );
46 //                else alert( "Login Failed" );
47 //            }, "json"
48 //        );
49 //    });
50 //});
51 </script>
52 {% endblock %}
```

ui/templates/dashboard.html 機器情報一覧ページ

ソースコード 8.8: 機器情報一覧ページ

```
1  {%- extends "layout.html" %}  
2  {%- block maincontent %}  
3  <div class="container-fluid" style="padding:70px_10px;min-height:2000px">  
4      <h1> Summary of Devices</h1>  
5      <button id="update" class="btn btn-info btn-md"><span class="glyphicon glyphicon-repeat"></span>&nbsp;Reload</button>&nbsp;Last Reload:<span id="updatetime"></span>  
6      <button id="toggleview" class="btn btn-info btn-md pull-right">Small</button>  
7      <div class="list-group" id="list">  
8          <a href="#" id="plus" class="list-group-item list-group-item-action" data-toggle="modal" data-target="#moreinfo" data-whatever="add" ><span class="glyphicon glyphicon-plus"></span></a>  
9      </div>  
10     </div>  
11  
12  
13     <!-- for add and mod device-->  
14     <div class="modal_fade" id="moreinfo" tabindex="-1" >  
15         <div class="modal-dialog modal-md">  
16             <div class="modal-content">  
17                 <div class="modal-header">  
18                     <button type="button" class="close" data-dismiss="modal">&times;</button>  
19                     <h4 class="modal-title">Add Device</h4>  
20                 </div>  
21                 <div class="modal-body">  
22                     <form class="form-horizontal" >  
23                         <div class="control-group has-success has-feedback" >  
24                             <label class="control-label" for="DeviceID">  
25                                 DeviceID</label>  
26                             <div class="controls" >  
27                                 <input name="DeviceID" class="form-control" type="text" placeholder="Device-ID" required="" />  
28                                 <span class="glyphicon glyphicon-ok form-control-feedback" ></span>  
29                             </div>  
30                         <div class="control-group" >  
31                             <label for="DeviceName">Device-Name</label>  
32                             <div class="controls" >  
33                                 <input name="DeviceName" class="form-control" type="text" placeholder="Device_Name" required="" />  
34                             </div>  
35                         </div>  
36                         <div class="control-group" >  
37                             <label for="Description">Descriptption</label>  
38                             <div class="controls" >  
39                                 <textarea rows="10" name="Description" class="form-control" ></textarea>  
40                             </div>  
41                         </div>  
42                         <div class="control-group" >  
43                             <div class="controls" >  
44                                 <button class="btn btn-default" data-dismiss="modal">Cancel</button>  
45                                 <button class="btn btn-default" type="submit">Add</button>  
46                             </div>  
47                         </div>  
48                     </form>  
49                     </div>  
50                 </div>  
51             </div>  
52         </div>  
53  
54     <!-- for messagebox -->  
55     <div class="modal_fade" id="messagebox" tabindex="-1" >  
56         <div class="modal-dialog modal-sm" >
```

```

57     <div class="modal-content">
58         <div class="modal-header">
59             <button type="button" class="close" data-dismiss="modal">&
60                 times;</button>
61             <h4 class="modal-title"></h4>
62         </div>
63         <div class="modal-body">
64         </div>
65         <div class="modal-footer">
66             <button name="yesbutton" class="btn btn-default">Yes</
67                 button>
68             <button name="nobutton" class="btn btn-default">No</button
69                 >
70         </div>
71     </div>
72 <div style="display:none">
73     <a href="#" id="listitem" class="list-group-item list-group-item-action listitems"
74         >
75             <h4 class="list-group-item-heading smallshow">
76                 <span name="DeviceName"></span><span name="icon" class="
77                     glyphicon glyphicon-heart pull-right"></span>
78             </h4>
79             <h3 class="list-group-item-heading normalshow">
80                 <span name="DeviceName"></span>
81             </h3>
82             <h3 class="list-group-item-heading moreshow">
83                 <span name="DeviceName"></span>
84                 <button name="edit" class="btn btn-default pull-right" ><span
85                     class="glyphicon glyphicon-edit"></span></button>
86                 <button name="remove" class="btn btn-default pull-right"><span
87                     class="glyphicon glyphicon-trash"></span></button>
88             </h3>
89             <div class="list-group-item-text normalshow" style="text-align:center">
90                 <span name="Status"></span><br />
91                 @<span name="HeartTime"></span>
92             </div>
93             <div class="list-group-item-text moreshow">
94                 DeviceID:&nbsp;<span name="DeviceID"></span><br />
95                 Now Status:&nbsp;<span name="Status"></span>&nbsp;(At <span name
96                     ="HeartTime"></span>)<br />
97                 Description:<br />
98                 <p name="Description">
99                 </p>
100                Commands:
101                <button class="btn btn-default" name="reboot"><span class="
102                    glyphicon glyphicon-refresh"></span></button>
103                <button class="btn btn-default" name="shutdown"><span class="
104                    glyphicon glyphicon-off"></span></button>
105            </div>
106        </a>
107    </div>
108 {%- endblock %}
109
110
111
112
113
114
115
116
117
118
119
120

```

```

121     for( var i in devlist ){
122         id = devlist[i].ID;
123         if( $("#" + id ).length ){//Update item
124             $("#" + id ).removeClass( "list-group-item-
125                 success_list-group-item-warning_list-group-
126                 -item-danger" );
127             if( devlist[i].Stat == "OK" ) $("#" + id ).addClass( "
128                 list-group-item-success" );
129             else if( devlist[i].Stat == "UP" ) $("#" + id ).addClass(
130                 "list-group-item-warning" );
131             else if( devlist[i].Stat == "NG" ) $("#" + id ).addClass(
132                 "list-group-item-danger" );
133             $("#" + id ).find( "span[name='DeviceName']" ).empty().append(
134                 devlist[i].Name );
135             $("#" + id ).find( "span[name='Status']" ).empty().append(
136                 devlist[i].Stat );
137             $("#" + id ).find( "span[name='HeartTime']" ).empty().append(
138                 (devlist[i].time==0)?"No_Data":devlist[i].time );
139             $("#" + id ).find( "p[name='Description']" ).empty().append(
140                 devlist[i].Description );
141             if( devlist[i].Stat != "OK" ) $("#" + id ).find( "span[
142                 name='icon']" ).removeClass( "glyphicon-
143                 heart" ).addClass( "glyphicon-alert" );
144             else $("#" + id ).find( "span[name='icon']" ).removeClass(
145                 "glyphicon-alert" ).addClass( "glyphicon-
146                 heart" );
147             }else{//Append item
148                 var item = $("#" + "#listitem" ).clone( true );
149                 var str = "list-group-item-";
150                 item.attr( "id", id );
151                 if( devlist[i].Stat == "OK" ) str += "success";
152                 else if( devlist[i].Stat == "UP" ) str += "warning";
153                 else if( devlist[i].Stat == "NG" ) str += "danger";
154                 item.addClass( str );
155                 item.find( "span[name='DeviceName']" ).append(
156                     devlist[i].Name );
157                 item.find( "button[name='edit']" ).attr( "data-
158                     whatever", devlist[i].ID );
159                 item.find( "span[name='DeviceID']" ).append(
160                     devlist[i].ID );
161                 item.find( "span[name='Status']" ).append( devlist[
162                     i].Stat );
163                 item.find( "span[name='HeartTime']" ).append(
164                     (devlist[i].time==0)?"No_Data":devlist[i].time );
165                 item.find( "p[name='Description']" ).append(
166                     devlist[i].Description );
167                 if( devlist[i].Stat != "OK" ) item.find( "span[name='
168                     icon']" ).removeClass( "glyphicon-heart" ).addClass(
169                     "glyphicon-alert" );
170                 else item.find( "span[name='icon']" ).removeClass(
171                     "glyphicon-alert" ).addClass( "glyphicon-
172                     heart" );
173             }
174             $("#" + "#list" ).append( item );
175             flag = true
176         }
177         if( flag ) $("#" + "#plus").appendTo('#list');
178     }
179     $("#" + "#updatetime" ).empty().append( data.Time );
180 },
181 error: function(){
182     alert( "HTTP_request_fail." );
183 },
184 complete: function(){
185     $("#" + "#update" ).prop( "disabled", false );
186 }
187 });
188 }
189
190 update();
191 setInterval( update, 60000 );
192 $("#" + "#update" ).click( update );
193
194 var view = ".normalshow";

```

```

171     var viewo = [240,100];
172     $("a>.smallshow, a>.moreshow").hide();
173     $("#listitem").click(function(){
174         if( $( this ).find(".moreshow").is( ":visible" ) ){
175             $( this ).find(".moreshow").hide();
176             $( this ).find( view ).fadeIn("slow");
177             $( this ).animate( { width:viewo[0],height:viewo[1] }, 200 );
178         }else{
179             $( this ).find(".smallshow,.normalshow").hide();
180             $( this ).find( ".moreshow" ).fadeIn("slow");
181             $( this ).animate( { width: "360px", height:"360px" }, 200 );
182         }
183     });
184     $("#toggleview" ).click( function(){
185         $(".listitems>div.moreshow:visible" ).parent().click();
186         $(".listitems" ).find( view ).hide();
187         if( view == ".normalshow" ){
188             view = ".smallshow";
189             viewo = [180,50];
190             $(this).html( "Medium" );
191         }else{
192             view = ".normalshow";
193             viewo = [240,100];
194             $(this).html( "Small" );
195         }
196         $(".listitems" ).find( view ).show();
197         $(".listitems" ).animate( {width:viewo[0],height:viewo[1]}, 200 );
198     });
199
200     $("#listitem_button[name='remove']").click( function( event ){
201         var ID = $(this).parent().parent().find( "span[name='DeviceID']" ).html();
202         $("#messagebox" ).find( ".modal-title" ).html( "Confirmation" );
203         $("#messagebox" ).find( ".modal-body" ).html( "Are you really want to remove this device?" );
204         $("#messagebox" ).find( "button[name='yesbutton']" ).off( "click.msg" ).on( "click.msg", function(){
205             $.ajax({
206                 type: "delete",
207                 url: moddev+ID,
208                 dataType: "json",
209                 success: function(data){
210                     $("#" + ID ).fadeOut("slow",function(){ $("#" + ID ).remove();
211                         ()});}
212                     $("#messagebox" ).modal("hide");
213                 },
214                 error: function(){
215                     alert( "HTTP request fail." );
216                 }
217             });
218         $("#messagebox" ).find( "button[name='nobutton']" ).off( "click.msg" ).on( "click.msg", function(){($("#messagebox").modal("hide");});
219         $("#messagebox" ).modal();
220         event.stopPropagation();
221     });
222
223     $("#listitem_button[name='edit']").click( function( event ){
224         var ID = $( this ).parent().parent().find( "span[name='DeviceID']" ).html();
225         var name = $( this ).parent().parent().find( "span[name='DeviceName']" ).html();
226         var desc = $( this ).parent().parent().find( "p[name='Description']" ).html();
227         $("#moreinfo_input[name='DeviceID']").val( ID );
228         $("#moreinfo_input[name='DeviceName']").val( name );
229         $("#moreinfo_textarea[name='Description']").val( desc.trim() );
230         $("#moreinfo").modal();
231         event.stopPropagation();
232     });
233     $("#moreinfo").on( "show.bs.modal", function(event){
234         var tmp = $(event.relatedTarget).data('whatever');
235         if( tmp == 'add' ){
236             $.ajax({
237                 type: "get",
238                 url: devid,
239                 dataType: "json",
240                 success: function( data ){
```

```

241                     $( "#moreinfo input[name='DeviceID']" ).val( data.ID );
242                 },
243                 error: function(){
244                     alert( "HTTP\u2022request\u2022fail." );
245                 }
246             });
247             $(this).find( "input [name='DeviceName']" ).val("");
248             $(this).find( "textarea[name='Description']" ).val("");
249             $(this).find( "input [name='DeviceID']" ).prop( "disabled", false );
250             $(this).find( ".glyphicon" ).show();
251             $(this).find( ".control-group:first" ).addClass( "has-success" );
252             $(this).find( "button[type='submit']" ).html( "Add" );
253             $(this).find( ".modal-title" ).html( "New\u2022Device" );
254         }else{
255             $(this).find( ".glyphicon" ).hide();
256             $(this).find( ".control-group:first" ).removeClass("has-success");
257             $(this).find( "input[name='DeviceID']" ).prop( "disabled", true );
258             $(this).find( "button[type='submit']" ).html( "Submit" );
259             $(this).find( ".modal-title" ).html( "Device\u2022"+$(this).find("input[name='DeviceName']").val() );
260         }
261     });
262     function checkID(){
263         var tmp = { "ID": $( "#moreinfo input[name='DeviceID']" ).val() };
264         if( tmp.length == 0 ) return;
265         $.ajax({
266             type: "post",
267             url: devid,
268             data: JSON.stringify( tmp ),
269             contentType: "application/json",
270             dataType: "json",
271             success: function(data){
272                 if( data.stat == "OK" ){
273                     $( "#moreinfo .has-feedback" ).removeClass( "has-error" )
274                         .addClass( "has-success" );
275                     $( "#moreinfo .form-control-feedback" ).removeClass("glyphicon-remove").addClass("glyphicon-ok");
276                 }else{
277                     $( "#moreinfo .has-feedback" ).removeClass( "has-success" )
278                         .addClass( "has-error" );
279                     $( "#moreinfo .form-control-feedback" ).removeClass("glyphicon-ok").addClass("glyphicon-remove");
280                 }
281             },
282             error: function(){
283                 alert( "HTTP\u2022request\u2022fail." );
284             }
285         });
286         $( "#moreinfo input[name='DeviceID']" ).change( checkID );
287         $( "#moreinfo input[name='DeviceID']" ).bind( "paste", function(){ setTimeout( checkID, 20 ); });
288         $( "#moreinfo form" ).submit( function(){
289             var DeviceID = $( "#moreinfo input[name='DeviceID']" ).val();
290             var DeviceName = $( "#moreinfo input[name='DeviceName']" ).val();
291             var Description = $( "#moreinfo textarea[name='Description']" ).val();
292             var tmp = { ID:DeviceID, Name:DeviceName, Description:Description };
293             if( $( "#moreinfo input[name='DeviceID']" ).prop("disabled") ) tmp.stat = "mod";
294             else tmp.stat = "new";
295             $.ajax({
296                 type: "post",
297                 url: moddev+DeviceID,
298                 data: JSON.stringify( tmp ),
299                 contentType: "application/json",
300                 dataType: "json",
301                 success: function( data ){
302                     if( data.stat == "OK" ){
303                         update();
304                         $( "#moreinfo" ).modal("hide");
305                     }else{
306                         alert( "Fail..." );
307                     }
308                 },
309                 error: function(){

```

```

309             alert( "HTTP\u00a5request\u00a5fail." );
310         }
311     });
312     return false;
313 });
314 $(".listitems button[name='shutdown']").click( function(event){
315     var DeviceID = $(this).parent().find( "span[name='DeviceID']" ).html();
316     $.ajax({
317         type: "post",
318         url: moddev+DeviceID+"/power",
319         data: JSON.stringify( {command:"shutdown"} ),
320         contentType: "application/json",
321         dataType: "json",
322         success: function( data ){
323             if( data.stat == "OK" ){
324                 update();
325                 $( this ).parent().children( "button" ).prop( "disabled", true
326             );
327             }else{
328                 alert( "Please\u00a5retry..." );
329             }
330         },
331         error: function(){
332             alert( "HTTP\u00a5request\u00a5fail." );
333         }
334     });
335     event.stopPropagation();
336 });
337 $(".listitems button[name='reboot']").click( function(event){
338     var DeviceID = $(this).parent().find( "span[name='DeviceID']" ).html();
339     $.ajax({
340         type: "post",
341         url: moddev+DeviceID+"/power",
342         data: JSON.stringify( { command:"reboot" } ),
343         contentType: "application/json",
344         dataType: "json",
345         success: function( data ){
346             if( data.stat == "OK" ){
347                 update();
348             }else{
349                 alert( "Please\u00a5retry..." );
350             }
351         },
352         error: function(){
353             alert( "HTTP\u00a5request\u00a5fail." );
354         }
355     });
356     event.stopPropagation();
357 });
358 </script>
359 {% endblock %}
360
361
362
363 {% block style %}
364 <style>
365 .list-group>a {
366     float:left;
367     margin:10px;
368     border-radius: 5px;
369 }
370 .list-group>a>p>span{
371     font-size: 2em;
372 }
373 .listitems {
374     width:240px;
375     height:100px;
376 }
377 </style>
378 {% endblock %}

```

ソースコード 8.9: 過去の機器状態一覧ページ

```

1  {%- extends "layout.html" %} 
2  {%- block maincontent %} 
3  <div class="container-fluid" style="padding:70px\10px;min-height:2000px">
4      <h1>Devices Log</h1>
5      <button id="update" class="btn\btn-info\btn-md"><span class="glyphicon\glyphicon-repeat"></span>&nbsp;Reload</button>&nbsp;Last Reload:<span id="updatetime"></span>
6      <button id="toggleview" class="btn\btn-info\btn-md\pull-right">Small</button>
7      <ol class="list-group" id="list">
8          </ol>
9      </div>
10
11     <div style="display:none">
12         <li href="#" id="listitem" class="list-group-item\clearfix\col-md-12">
13             <div style="float:left" class="col-md-2\col-sm-3\col-xs-4\normalshow">
14                 Name:&nbsp;<span name="DeviceName"></span><br />
15                 ID:&nbsp;<span name="DeviceID"></span><br />
16                 Status:&nbsp;<span name="Status"></span>(<span name="HeartTime">
17                     ></span>)<br />
18                 Uptime:&nbsp;<span name="Uptime">xx day xx:xx:xx </span><br />
19             <div style="float:left;\display:none" class="col-md-2\col-sm-3\col-xs-4\smallshow">
20                 <span name="DeviceName"></span>(<span name="DeviceID"></span>)
21             </div>
22             <div style="float:right" class="col-md-10\col-sm-9\col-xs-8">
23                 <textarea name="log"></textarea>
24             </div>
25         </li>
26     </div>
27     {%- endblock %}
28
29
30
31     {% block script %}
32     <script>
33     Date.prototype.toString = function(){
34         var y = this.getFullYear();
35         var m = this.getMonth() + 1;
36         var d = this.getDate();
37         var H = this.getHours();
38         var M = this.getMinutes();
39         var S = this.getSeconds();
40         if( m < 10 ) m = "0" + m;
41         if( d < 10 ) d = "0" + d;
42         if( H < 10 ) H = "0" + H;
43         if( M < 10 ) M = "0" + M;
44         if( S < 10 ) S = "0" + S;
45         return y+"/"+m+"/"+d+"_"+H+"_"+M+":_"+S;
46     }
47     $(function(){
48         function update(){
49             $("#update").prop( "disabled", true );
50             $.ajax({
51                 type: "get",
52                 url: "{{url_for("devicestatus_all")}}",
53                 dataType: "json",
54                 success: function( data ){
55                     var dl = data.devices;
56                     for( var i in dl ){
57                         var id = dl[i].ID;
58                         if( $('#'+id).length ){ //Update Item
59                             $('#'+id).find("span[name='DeviceName']").html(
60                                 dl[i].Name );
61                             $('#'+id).find("span[name='DeviceID']").html( dl
62                                 [i].ID );
63                             $('#'+id).find("span[name='Status']").html( dl[i].

```

```

        Status );
    $( '#'+id ).find("span[name='HeartTime']").html(
        dh[i].time );
} else{ //Append Item
    var item = $( "#listitem" ).clone( true );
    item.attr( "id", dl[i].ID );
    item.find( "span[name='DeviceName']" ).html( dl[i].
        Name );
    item.find( "span[name='DeviceID']" ).html( dl[i].ID
        );
    item.find( "span[name='Status']" ).html( dl[i].
        Status );
    item.find( "span[name='HeartTime']" ).html( (dl[i].
        time==0?"NoData":dl[i].time) );
    $( "#list" ).append( item );
}
$.ajax({
    type: "get",
    url: "{url_for('devicelog', DeviceID='')}",
    replace('/','/' + id + '/'),
    dataType: "json",
    success: function( data ){
        var txt = "";
        for( i in data.data ){
            var st = new Date( data.data[i].stime
                * 1000 );
            var et = new Date( data.data[i].
                etime * 1000 );
            txt += st.toString() + " ~ " + et.
                toString() + "\n";
            switch( data.data[i].stat ){
                case 1:
                    txt += "NC";
                    break;
                case 2:
                    txt += "OK";
                    break;
                default:
                    txt += "NG";
                    break;
            }
            txt += "\n";
        }
        $( "#"+data.ID+" textarea[name='log']"
            ).val( txt );
    },
    error: function(){
        alert("HTTP request fail...");
    }
});
},
error: function(){
    alert( "HTTP request fail..." );
}
);
update();
});
var chart = c3.generate({
    data: {
        x:"x",
        xFormat: '%Y-%m-%d %H:%M:%S',
        columns: [
            [
                'x',
                '2013-01-01 00:10:00',
                '2013-01-01 00:11:00',
                '2013-01-01 00:12:00',
                '2013-01-01 00:13:00',
                '2013-01-01 00:14:00',
                '2013-01-01 00:15:00'
            ],

```

```

125      [ 'data2', 2,2,0,1,1,2]
126    ],
127  },
128  axis: {
129    x: { type: 'timeseries', tick:{ format: "%Y-%m-%d %H:%M:%S" } },
130    y: { max:2, tick:{ format: function(d){
131      switch( d ){
132        case 0: return "NG";
133        case 1: return "NC";
134        case 2: return "OK";
135        default: return "";
136      }
137    },
138    values:[0,1,2]
139  } }
140  }
141 });
142 </script>
143 <!--script>
144 var devstaturl = "{{url_for('devicestatus_all')}}";
145 var devid = "{{url_for('gen_device_id')}}";
146 var moddev = "{{url_for('mod_device',DeviceID='')}}";
147 $(function(){
148   function update(){
149     $("#update").prop( "disabled", true );
150     $.ajax({
151       type: "get",
152       url: devstaturl,
153       dataType: "json",
154       success: function( data ){
155         var devlist = data.devices;
156         var flag = false
157         for( var i in devlist ){
158           id = devlist[i].ID;
159           if( $("#" + id ).length ){//Update item
160             $("#" + id ).removeClass( "list-group-item-success" );
161             $("#" + id ).addClass( "list-group-item-warning" );
162             if( devlist[i].Stat == "OK" ) $("#" + id ).addClass( "list-group-item-success" );
163             else if( devlist[i].Stat == "UP" ) $("#" + id ).addClass( "list-group-item-warning" );
164             else if( devlist[i].Stat == "NG" ) $("#" + id ).addClass( "list-group-item-danger" );
165             $("#" + id ).find( "span[name='DeviceName']" ).empty().append( devlist[i].Name );
166             $("#" + id ).find( "span[name='Status']" ).empty().append( devlist[i].Stat );
167             $("#" + id ).find( "span[name='HeartTime']" ).empty().append( (devlist[i].time==0)?"No Data":devlist[i].time );
168             $("#" + id ).find( "p[name='Description']" ).empty().append( devlist[i].Description );
169             if( devlist[i].Stat != "OK" ) $("#" + id ).find( "span[name='icon']" ).removeClass( "glyphicon-heart" ).addClass( "glyphicon-alert" );
170             else $("#" + id ).find( "span[name='icon']" ).removeClass( "glyphicon-alert" ).addClass( "glyphicon-heart" );
171           }else{//Append item
172             var item = $( "#listitem" ).clone( true );
173             var str = "list-group-item-";
174             item.attr( "id", id );
175             if( devlist[i].Stat == "OK" ) str += "success";
176             else if( devlist[i].Stat == "UP" ) str += "warning";
177             else if( devlist[i].Stat == "NG" ) str += "danger";
178             item.addClass( str );
179             item.find( "span[name='DeviceName']" ).append( devlist[i].Name );
180             item.find( "button[name='edit']" ).attr( "data-whatever", devlist[i].ID );
181             item.find( "span[name='DeviceID']" ).append( devlist[i].ID );

```

```

182         item.find( "span[name='Status']" ).append( devlist[
183             i].Stat );
184         item.find( "span[name='HeartTime']" ).append( (
185             devlist[i].time==0?"No_Data":devlist[i].time );
186         item.find( "p[name='Description']" ).append(
187             devlist[i].Description );
188         if( devlist[i].Stat != "OK" ) item.find( "span[name='
189             icon']" ).removeClass( "glyphicon-heart" );
190             addClass( "glyphicon-alert" );
191         else item.find( "span[name='icon']" ).removeClass(
192             "glyphicon-alert" ).addClass( "glyphicon-
193             heart" );
194         $( "#list" ).append( item );
195         flag = true
196     }
197     if( flag ) $("#plus").appendTo('#list');
198     $( "#updatetime" ).empty().append( data.Time );
199 },
200 error: function(){
201     alert( "HTTP request fail." );
202 },
203 complete: function(){
204     $("#update" ).prop( "disabled", false );
205 }
206 });
207
208 update();
209 setInterval( update, 60000 );
210 $( "#update" ).click( update );
211
212 var view = ".normalshow";
213 var viewo = [240,100];
214 $( "a>.smallshow, a>.moreshow" ).hide();
215 $( "#listitem" ).click(function(){
216     if( $( this ).find(".moreshow").is( ":visible" ) ){
217         $( this ).find(".moreshow").hide();
218         $( this ).find( view ).fadeIn("slow");
219         $( this ).animate( { width:viewo[0],height:viewo[1] }, 200 );
220     }else{
221         $( this ).find(".smallshow,.normalshow").hide();
222         $( this ).find( ".moreshow" ).fadeIn("slow");
223         $( this ).animate( { width: "360px", height:"360px" }, 200 );
224     }
225 });
226 $( "#toggleview" ).click( function(){
227     $(".listitems>div.moreshow:visible" ).parent().click();
228     $(".listitems" ).find( view ).hide();
229     if( view == ".normalshow" ){
230         view = ".smallshow";
231         viewo = [180,50];
232         $(this).html( "Medium" );
233     }else{
234         view = ".normalshow";
235         viewo = [240,100];
236         $(this).html( "Small" );
237     }
238     $(".listitems" ).find( view ).show();
239     $(".listitems" ).animate( {width:viewo[0],height:viewo[1]}, 200 );
240 });
241
242 $( "#listitem_button[name='remove']" ).click( function( event ){
243     var ID = $(this).parent().parent().find( "span[name='DeviceID']" ).html();
244     $( "#messagebox" ).find( ".modal-title" ).html( "Confirmation" );
245     $( "#messagebox" ).find( ".modal-body" ).html( "Are you really want to remove
246         this device?" );
247     $( "#messagebox" ).find( "button[name='yesbutton']" ).off( "click.msg" ).on( "
248         click.msg", function(){
249             $.ajax({
250                 type: "delete",
251                 url: moddev+ID,
252                 dataType: "json",
253                 success: function(data){

```

```

247      $( "#"+ID ).fadeOut("slow",function(){$("#"+ID).remove()
248          ());
249      $( "#messagebox" ).modal("hide");
250  },
251  error: function(){
252      alert( "HTTP\u2022request\u2022fail." );
253  }
254 });
255 $( "#messagebox" ).find( "button[name='nobutton']" ).off( "click.msg" ).on( "click.msg",
256     function(){($("#messagebox").modal("hide");});
257     $( "#messagebox" ).modal();
258     event.stopPropagation();
259 });
260 $( "#listitem\u2022button[name='edit']" ).click( function( event ){
261     var ID = $( this ).parent().parent().find( "span[name='DeviceID']" ).html();
262     var name = $( this ).parent().parent().find( "span[name='DeviceName']" ).html();
263     var desc = $( this ).parent().parent().find( "p[name='Description']" ).html();
264     $( "#moreinfo\u2022input[name='DeviceID']" ).val( ID );
265     $( "#moreinfo\u2022input[name='DeviceName']" ).val( name );
266     $( "#moreinfo\u2022textarea[name='Description']" ).val( desc.trim() );
267     $( "#moreinfo" ).modal();
268     event.stopPropagation();
269 });
270 $( "#moreinfo" ).on( "show.bs.modal", function(event){
271     var tmp = $(event.relatedTarget).data('whatever');
272     if( tmp == 'add' ){
273         $.ajax({
274             type: "get",
275             url: devid,
276             dataType: "json",
277             success: function( data ){
278                 $( "#moreinfo\u2022input[name='DeviceID']" ).val( data.ID );
279             },
280             error: function(){
281                 alert( "HTTP\u2022request\u2022fail." );
282             }
283         });
284         $(this).find( "input[name='DeviceName']" ).val("");
285         $(this).find( "textarea[name='Description']" ).val("");
286         $(this).find( "input[name='DeviceID']" ).prop( "disabled", false );
287         $(this).find( ".glyphicon" ).show();
288         $(this).find( ".control-group:first" ).addClass( "has-success" );
289         $(this).find( "button[type='submit']" ).html( "Add" );
290         $(this).find( ".modal-title" ).html( "New\u2022Device" );
291     }else{
292         $(this).find( ".glyphicon" ).hide();
293         $(this).find( ".control-group:first" ).removeClass("has-success");
294         $(this).find( "input[name='DeviceID']" ).prop( "disabled", true );
295         $(this).find( "button[type='submit']" ).html( "Submit" );
296         $(this).find( ".modal-title" ).html( "Device\u2022"+$(this).find("input[name='DeviceName']" ).val() );
297     }
298 });
299 function checkID(){
300     var tmp = { "ID": $( "#moreinfo\u2022input[name='DeviceID']" ).val() };
301     if( tmp.length == 0 ) return;
302     $.ajax({
303         type: "post",
304         url: devid,
305         data: JSON.stringify( tmp ),
306         contentType: "application/json",
307         dataType: "json",
308         success: function(data){
309             if( data.stat == "OK" ){
310                 $( "#moreinfo\u2022.has-feedback" ).removeClass( "has-error" )
311                     .addClass( "has-success" );
312                 $( "#moreinfo\u2022.form-control-feedback" ).removeClass("glyphicon-remove")
313                     .addClass("glyphicon-ok");
314             }else{
315                 $( "#moreinfo\u2022.has-feedback" ).removeClass( "has-success" )
316                     .addClass( "has-error" );
317                 $( "#moreinfo\u2022.form-control-feedback" ).removeClass(" glyphicon-ok")
318                     .addClass("glyphicon-remove");
319             }
320         }
321     });
322 }

```

```

            glyphicon-ok").addClass("glyphicon-remove");
315        }
316    },
317    error: function(){
318        alert( "HTTP\u2022request\u2022fail." );
319    }
320 });
321 }
322 $("#moreinfo\u2022input[name='DeviceID']").change( checkID );
323 $("#moreinfo\u2022input[name='DeviceID']").bind( "paste", function(){ setTimeout( checkID,
324 20 ); });
325 $("#moreinfo\u2022form").submit( function(){
326     var DeviceID = $("#moreinfo\u2022input[name='DeviceID']").val();
327     var DeviceName = $( "#moreinfo\u2022input[name='DeviceName']" ).val();
328     var Description = $( "#moreinfo\u2022textarea[name='Description']" ).val();
329     var tmp = { ID:DeviceID, Name:DeviceName, Description:Description };
330     if( $("#moreinfo\u2022input[name='DeviceID']").prop("disabled") ) tmp.stat = "mod";
331     else tmp.stat = "new";
332     $.ajax({
333         type: "post",
334         url: moddev+DeviceID,
335         data: JSON.stringify( tmp ),
336         contentType: "application/json",
337         dataType: "json",
338         success: function( data ){
339             if( data.stat == "OK" ){
340                 update();
341                 $("#moreinfo").modal("hide");
342             }else{
343                 alert( "Fail..." );
344             }
345             error: function(){
346                 alert( "HTTP\u2022request\u2022fail." );
347             }
348         });
349         return false;
350     });
351 $("#listitem\u2022button[name='shutdown']").click( function(){
352     $.ajax({
353         type: "post",
354         url: moddev+DeviceID+"/power",
355         data: JSON.stringify( {command:"shutdown"} ),
356         contentType: "application/json",
357         dataType: "json",
358         success: function( data ){
359             if( data.stat == "OK" ){
360                 update();
361                 $( this ).parent().children( "button" ).prop( "disabled", true
362 );
363             }else{
364                 alert( "Please\u2022retry..." );
365             }
366             error: function(){
367                 alert( "HTTP\u2022request\u2022fail." );
368             }
369         });
370     });
371 $("#listitem\u2022button[name='reboot']").click( function(){
372     $.ajax({
373         type: "post",
374         url: moddev+DeviceID+"/power",
375         data: JSON.stringify( { command:"reboot" } ),
376         contentType: "application/json",
377         dataType: "json",
378         success: function( data ){
379             if( data.stat == "OK" ){
380                 update();
381             }else{
382                 alert( "Please\u2022retry..." );
383             }
384             error: function(){

```

```
386                     alert( "HTTP\u00d7request\u00d7fail." );
387                 });
388             });
389         });
390     </script-->
391 {%
392 % endblock%
393
394
395 {% block style %}
396 <style>
397 ol > li.list-group-item {
398     margin: 10px 0px;
399 }
400 </style>
401 {%
402 % endblock %}
```
