

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 虹槻

指導教員

横山 輝明

提出日

2017 年 1 月 28 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
第 2 章	既存の解決策とその課題	2
2.1	IoT サービスの構造の分析	2
2.2	機器設置箇所に行って、直接確認する	2
2.3	ICMP Ping を活用する	2
2.4	SNMP を利用する	3
2.5	Zabbix を使用する	3
2.6	Fluentd Elasticsearch Kibana を利用する	3
2.7	Telegraf Influxdb Grafana を利用する	3
2.8	SORACOM Air を利用する	3
2.9	VPN を利用する	4
第 3 章	提案する解決策	5
3.1	岡本商店街での事例	5
3.1.1	実験概要	5
3.1.2	課題と考察	5
3.2	学内での wifi を用いた人流観測	6
3.2.1	課題と考察	6
3.3	要件の抽出	6
第 4 章	設計と実装	7
4.1	想定するユーザーの定義	7
4.2	課題と機能の関係	7
4.3	システムの構成	7
4.4	構成部品のそれぞれの動き	9
4.4.1	監視エージェントの動き	9
4.4.2	サービスの動き	9
4.5	ユーザーの動き？	9
第 5 章	検証と考察	11
5.1	実験概要	11
5.1.1	実験目的	11
5.1.2	実験方法	11
5.2	準備	11
5.3	経過	11
5.4	考察	11
第 6 章	結論	12

第 7 章 謝辞	13
第 8 章 参考文献	14

内容梗概

近年、半導体技術の進歩により、コンピューターの小型化・低価格化が進んでいる。また、インターネット回線網の普及もあり、Internet of Things という概念が注目され、それによって収益を得る IoT サービスが登場してきた。Internet of Things(IoT) とは、様々な物がインターネットにつながり、相互に情報を交換し合うことで、様々な自動化を実現する概念である。

しかし、IoT サービスを開発・運用するには、開発コストの問題・セキュリティの問題・稼働率の問題など様々な問題がある。

そこで、本研究では、IoT デバイスの死活監視問題に焦点を当て、IoT サービスとは独立した IoT デバイスの監視サービスを開発することにより、デバイスの故障検知に係る問題の解決を図ることにした。システムの構築に先立って、どのような機能が必要となるのか、実験し、デバイスの電源の状態(電源が入っているのか・入っていないのか)・ネットワークの状態(インターネットへ接続されているのかいないのか)が時系列に沿って整理されている事で、対処が決まる事が分かった。そこで、上記必要な機能を実装したシステムを提供し、協力者の理解を得て検証し評価を得た。

第1章 はじめに

(IoT とは何なのか) 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 サービスとは何なのか、何故期待されているのかを説明) 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 機器とサーバー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。サーバーは、IoT 機器からの情報を蓄積・分析し、IoT 機器へ指示を送るか、ユーザーへ分析結果を表示する機能を持つこれら IoT 機器とサーバーが連携することで、IoT サービスは利便性をユーザーへ提供している。

(IoT サービスの提供にはどのような事が必要となるのか) IoT サービスを円滑に提供するには、IoT 機器とサーバー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 機器の動作状態や通信状態の監視が、何故困難なのか) ところが、数も多く、さまざまなネットワークを介して接続される IoT 機器の監視は困難な問題である。IoT 機器が設置される様々なネットワークを全て把握する事は、IoT 機器が多量であることを考えると現実的ではない。多量の IoT 機器を個々に識別し、異常を検知することも難しい。

(それらを解決する為に、何が求められるのか) そのため、設置されるネットワークに関係なく状態が監視できることが求められる。また、IoT 機器の状態を一覧して確認できることや、IoT 機器の過去の動作状態や通信状態を確認できることが必要である。

(何を提案するのか) そこで、我々は、IoT 機器からの通知に基づいた機器監視サービスを提案する。IoT 機器が自身の過去の動作状態や通信状態を記録し、機器監視サービスがそれら記録を可視化する。この仕組みを用いることで、IoT 機器が設置されるネットワークに関係無く状態を監視することや、IoT 機器の過去の状態や通信状態を確認することを容易にする。本研究では、IoT 機器からの通知による機器の設置環境によらない機器の監視を行うことにより、IoT サービスの維持を容易にするシステムの開発に取り組む。

(論文の構成) 本論文では、IoT 機器の監視困難の問題を取り上げ、その問題解決のための監視サービスを開発し、効果を報告する。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 機器監視サービスの機能要件について述べる。第4章では、IoT 機器監視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 既存の解決策とその課題

序論で述べたとおり、本研究で解決する問題は以下の3つである。

- 設置したものが正常に稼働し始めたかどうか確認するのが面倒
設置者が、機器の操作を知っている必要が有る。
また、ディスプレイ等をつけないことが多いので、別途確認する手段（ディスプレイとキーボードを持参等）を用意する必要がある。
- 設置後、正常に稼働しているのか確認するのが面倒
NAPT の内側に設置されている事が多いので、Ping や snmp では確認できない。
また、ネットワークの断絶等があった場合、稼働状況を確認できない。
- いつ稼働していていつ稼働していなかったのか管理するのが大変
いつ稼働していていつ稼働していなかったのかがわからないと、データを正確に分析する事が出来ない。

2.1 IoT サービスの構造の分析

ここで、対象となる IoT サービスの構造について説明する。IoT サービスは、IoT 機器（機器）、ネットワーク、サーバーの3層から成り立っている。IoT 機器は、センシングしたデータを送り出す、または、送られてきたデータを元に何か動作を行うものである。ネットワークは、末端のネットワークと、インターネットとに別れる。末端のネットワークとインターネットとの出口には、NAPT が置かれていることが多い。サーバーは、IoT 機器から送られてきたデータを蓄積・分析し、表示または、IoT 機器へ通知する。

以下に状態監視のシステムを導入？しない場合の解決策を上げる

2.2 機器設置箇所に行って、直接確認する

直接設置箇所に行き、確認を行うという方法である。この場合、ディスプレイとキーボードを持参するなどする必要がある。そして、機器を良く知る者を行かせる必要もある。設置台数が多いことや、設置個所が離れていることから、あまり現実的ではない。

2.3 ICMP Ping を活用する

ICMP とは、InternetControlMessageProtocol の略であり、IP パケットの送り元から送り先への間で起きた問題を通知する役割を持つ。ICMP には、ICMP echo request と、ICMP 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 を返送しなければならない。Ping は ICMP echo を送信する為のプログラムで、IP 網のトラブルの発見の他、特定の IP アドレスを持つ機器が稼働しているかどうか確認するためにも使われている。Ping を使用して、IoT

機器の稼働確認を行うというのがこの解決策である。しかし、ICMP パケットは、セキュリティの都合上、ネットワーク機器で転送しないよう設定されている場合が多い。また、一般的なネットワークでは、インターネットとの接続点にてネットワークを分離している事がある。その場合、IoT 機器の IP アドレスは、IoT 機器の所属するネットワークでのみ通用するアドレスとなるため、外部から ICMP パケットを送ることは出来ない。

2.4 SNMP を利用する

SNMP とは、SimpleNetworkManagementProtocol の略で、ネットワークに接続された機器を監視・制御するために作られている。CPU の状態や起動してから時間、メモリ使用量等を取得できるが、設定がめんどくさい。また、制御もできるため、ネットワーク機器にてブロックされてしまう恐れがある。

しかし、これらの手法では、解決に至っていない。そこで、通常(?)は、次のような方法で解決を図っている。

2.5 Zabbix を使用する

2.6 Fluentd Elasticsearch Kibana を利用する

Fluentd とは、ログの転送を目的としたアプリケーション。ログを転送する他、場合によってはバッファリング等も行う。Elasticsearch とは、データベースの一つ。Kibana とは、可視化ツールの一つで、グラフの描画などが行える。しかし、それぞれが「そのための」プログラムなので、それらを使って可視化までこぎつけるには、それぞれに対して設定が必要と鳴る。また、Fluentd が動く機器である必要がある。更に、Kibana は可視化の為のシステムなので、一つの機器の状態を可視化するだけでも、ダッシュボードの作成からパネルの作成までする必要があり、多数の機器の状態を見るためには、それを複数回繰り返す必要がある。

2.7 Telegraf Influxdb Grafana を利用する

Influxdb とは時系列データベースの一つで、データを時系列に沿って整理し格納する。Telegraf とは Influxdb ヘメモリ使用量等を格納するためのエージェントプログラムである。Grafana とは可視化ツールの一つで、グラフの描画等を行う事ができる。これも、FluentdElasticSearchKibana の時のように、それぞれに対して設定が必要で、機器の上で Telegraf が動く必要がある。また、Grafana についても、Kibana と同様で、多数の機器を監視しようとするとても手間がかかる。

2.8 SORACOM Air を利用する

IoT 向けネットワークとして SORACOMAir というサービスがある。SORACOMAir は携帯電話回線を使用し、IoT 向けのネットワークを提供している。携帯電話回線を使用するため、SIM カードを用いており、SIM カードの ID により、個体の識別と通信状況の確認が行える。しかし、IoT 機器に SIM カードリーダー(?)等を接続する必要が有る。また、SORACOMAir では、インターネットとの接続に NAPT を使用しているため、外から状態を確認することは困難である。

2.9 VPNを利用する

VPNとは、VirtualPrivateNetworkの略で、インターネット越しに仮想的で閉鎖的なネットワークを構築するものである。しかし、VPNを利用するには、VPNクライアントが使用できる必要がある。また、仮想的なネットワークのIPアドレス帯とIoT機器がつながっているネットワークのIPアドレス帯が被らないように調節する必要があるため、IoT機器がつながるネットワークの構成を全て把握している必要がある。(IPアドレス帯がかぶると、どちらのIoT機器がどちらのインターフェースにパケットを送り出すべきなのかわからなくなるため、調節が必要)IoT機器の数を考えると、ネットワークの構成を把握することは困難である。

第3章 提案する解決策

そこで、新たに IoT 機器の稼働状態を監視することに特化したシステムを開発・提供することで、問題が解決できるのではないかと考えた。まず、システムの要件を抽出するために、以下のような実験を行った。

3.1 岡本商店街での事例

岡本商店街とは、兵庫県東灘区にある阪急岡本駅と JR 摂津本山駅の間にある商店街の事である。目的は、岡本商店街内を往来する人の流れを観測し、商店街の活性化につなげる事であったが、有用な知見（？）が得られたので、事例として上げることにする。

3.1.1 実験概要

スマートフォンの Wifi 接続機能を ON にした時、スマートフォンから定期的にプローブパケットというものが、送信される。そのプローブパケットを観測するためのソフトウェアとして、ampsence がある。本実験では、その ampsence と RaspberryPi を用いてプローブパケットを観測し、Web アプリケーションによって可視化することを行った。RaspberryPi は、岡本商店街内にある商店に設置させてもらい、インターネットとの接続は、商店に敷設済みの Wifi ネットワークを使用させてもらう他、1 台のみ SORACOM Air を使用した。SORACOM Air とは、IoT 向けデータ通信を提供するサービスで、携帯電話網を用いている。そのため、RaspberryPi に、SIM モジュールと SIM カードを接続しなければならなかった。

3.1.2 課題と考察

その実験の際、次のような問題があった。

- 機器が稼働していないことがあった
コンセントが抜け、機器が稼働していないことがあった。
そのため、分析する段階において、機器からログを回収し、全ての機器が稼働している時間帯を選び、分析する必要があった。
また、そのログのタイムスタンプが、UnixTime であったため、可読性が低く時間がかかった。
機器の稼働を監視していなかったため、機器が復旧するまでに必要以上に時間がかかり、分析の際、必要なデータが集まらないため、分析を諦めたものもあった。
- 迅速に稼働状態の確認が行えない
RaspberryPi を使用したセンサーであったため、ディスプレイやキーボードはついておらず、現地に行って確認するためにはディスプレイやキーボードを持っていく必要があった。
また、現地に行かなくてはならないのでとても手間であった。
そのため、遠隔からアクセスすることにした。

- 遠隔からのアクセスができない問題

しかし、遠隔からのプログラムの修正、稼働の監視を行う際、商店のネットワークと SORACOM Air のネットワークにはインターネットとの間に NAPT が入っており、外からアクセス不能だった。

VPN を利用しアクセスするが、稼働の確認毎に VPN サーバーにアクセスしてからの確認となるため、VPN に関する知識が必要だった。

また、簡単に稼働を確認できなかった。

- 機器の回収に手間がかかった

機器を回収する際、保存していたデータを破損しないために、正常にシャットダウンさせる必要があった。

そのため、回収作業の際もディスプレイとキーボードを持参し正常にシャットダウンさせてからコンセントを抜く等、手間がかかった。

上記より、機器の稼働状態を確認する事はとても重要であることが分かった。

3.2 学内での wifi を用いた人流観測

学内にて、wifi を用いた人流観測を行った。構成としては、先の事例での構成とほぼ同じものを学内でも行った。先の事例から得た問題点を踏まえて行った。

3.2.1 課題と考察

その際にも次のような問題があった。

- 分析の為に、機器の状態を常に監視している必要があった
岡本商店街での実験を踏まえ、機器の状態を定期的に監視し、機器トラブルに備えた。
- 機器がネットワークに接続できていない事があった
機器のトラブルにより、機器がネットワークに接続できていない事があった。
機器の様子を見に行った所、電源が入っているのに繋がっていないというトラブルがあった。

3.3 要件の抽出

上記の事例・実験から、以下のような機能が必要となることが分かった。

- IoT 機器の稼働状態がわかる
IoT 機器の稼働状態は、稼働している・稼働していない・ネットワークに接続されていないの 3 つ必要である。また、一覧で閲覧することが可能である事が望ましい。
- IoT 機器の稼働状態の記録を閲覧することができる
データの分析を行う際に、それら稼働状況の記録が必要になる事が分かった。
また、それらの記録が時刻と共に、整理されている必要があることも分かった。
- IoT 機器の停止・再起動を行うことができる
保存していたデータを破損しないために、IoT 機器を正常にシャットダウンさせる必要があり、その手順のために IoT 機器を回収することが手間であることが分かった。

そこで、IoT サービスとは独立した、IoT 機器の稼働状態を監視・管理することを簡単にするサービスを開発し、提供すれば良いのではないかと考えた。

第4章 設計と実装

実験から抽出した要件に基づいて、システムを作成した。要件は以下のとおりである

- IoT 機器の稼働状態が確認できる
機器の状態については、機器が稼働していない・機器が稼働している・機器が稼働しているがネットワークに繋がっていないの3つ。
- IoT 機器の稼働状態の記録を閲覧することができる
IoT 機器の稼働状態の記録が時系列に整理され、確認できることが必要である。
- IoT 機器の停止、再起動を行うことができる
回収に備えて、機器の停止、再起動を行う事ができると良い。

4.1 想定するユーザーの定義

ユーザーは、IoT サービスを構築しようとしている SI さんと定義する。

4.2 課題と機能の関係

本システムは各課題に対し、以下のような機能を用いて解決する。

- IoT 機器の稼働状態が簡単に確認できない
直近の IoT 機器から通知をすることにより、直近の IoT 機器の状態を取得し、Web 画面にて一覧表示することにより解決する。
- IoT 機器の稼働状態の記録を整理するのに困る
IoT 機器から定期的に通知をもらい、それをサーバー側で整理してから格納し、ユーザーが見やすいよう Web 画面で表示することで解決する。
- IoT 機器の停止、再起動がしにくい
IoT 機器からの通知の返答として、シャットダウンや再起動と言ったメッセージを渡すことで、機器の停止、再起動を可能にする。また、Web 画面からこれらを行えるよう工夫した。

4.3 システムの構成

システムの構成は以下のとおりである。本システムは、エージェント、サーバーの2つの要素で成り立っている。エージェントはIoT 機器にインストールされ、サーバーに対し定期的に通知を送る。また、サーバーは、エージェントからの通知を記録する。サーバーはユーザーへのインターフェースも提供しており、ユーザーはブラウザを使用して本システムを利用する。

- エージェントについて

- インストールされる機器
RaspberryPi(Raspbian jessie)・Intel Edison(yocto linux) にインストールされる。
 - 機能
システムに対し、現在の状態を定期的に通知する。
 - 構成要素
シェルスクリプト一つ (agent.sh)
- サーバーについて
 - 環境
Ubuntu16.04 (xenial)
 - インストールしたもの
Python3 Influxdb
 - 構成要素とその説明
 - * API サーバ
 - ・ 使用したライブラリ
Falcon
 - ・ 機能
エージェントから送られてきたデータを展開しログ用データベースに格納する。
 - ・ ソースコード
 - * ログ用データベース
 - ・ 使用したもの
Influxdb
 - ・ 機能と説明
ログを蓄積する。他にもデータベースとしての選択肢があったが、時系列に整理され、検索が早いらしいことから採用した。
 - * 機器情報用データベース
 - ・ 使用したもの
sqlite3
 - ・ 機能と説明
機器に関する情報を記録する。Python から使いやすかったので採用した。
 - * Web アプリケーション
 - ・ 使用したライブラリ
Flask Bootstrap JQuery
 - ・ 機能
ユーザーからの操作を受け付け、データベースに反映する。また、必要な情報をデータベースから取得し表示する。
 - ・ ソースコード

4.4 構成部品のそれぞれの動き

4.4.1 監視エージェントの動き

監視対象となる IoT 機器には、開発した監視エージェントが入っているものとし、起動時に自動でエージェントプログラムを起動するよう設定されているものとする。ログファイルの中には、過去のシーケンス番号が入っている。監視エージェントは、起動後、ログファイルがあるか確認し、あった場合、ログファイルから過去のシーケンス番号を読み出す。無かった場合、過去のシーケンス番号を 0 にする。その後、現在のシーケンス番号を 0 にする。監視エージェントは、監視サービスに対し過去のシーケンス番号と現在のシーケンス番号、自身が現在正常である旨のメッセージを送信する。監視サービスから返答があった場合、受理されたとみなし、現在のシーケンス番号を 0 にする。その後、過去のシーケンス番号とログファイルを削除する。また、返答に停止・再起動のコマンドが含まれていた場合、そのコマンドを実行し終了する。返答が無かった場合、ネットワークに障害があったとみなし、シーケンス番号をインクリメントし、ログファイルに保存する。この動きを約 1 分ごとに繰り返す。

4.4.2 サービスの動き

監視エージェントからメッセージを受け取った時の動き

監視エージェントからメッセージを受け取った場合、DB の末尾がコマンドを示すものであった場合、そのコマンドを変数に記録しておく。過去のシーケンス番号の数だけ、接続されていなかった旨を、最後に通信があった時刻から DB に格納し、現在のシーケンス番号の数だけ、接続されていなかった旨を、現在の時刻からさかのぼって DB に格納する。また、現在の時刻にて機器の状態が正常であった旨を DB に格納する。最後に、コマンドが格納されている変数の内容と、受理した旨のメッセージを監視エージェントに対し送信する。

ブラウザから入力を受けた場合の動き

まず、クッキーからセッションを読みだし、ログインしていないユーザーであった場合、ログインページへと誘導する。

- 現在の機器の状態の確認を求められた場合
サービスは、DB から、機器の現在の状態とその他の機器情報を取得し、返却する。
- 機器情報の追加・変更・削除を求められた場合
サービスは、DB へ変更を保存する。
- ログ一覧を求められた場合
サービスは、DB から機器の状態を取得・整理し、返却する

4.5 ユーザーの動き？

ユーザーの動きは以下のとおりである。

- 機器を追加するとき
 1. ブラウザからサービスにアクセスし、ログインする。

2. 画面から追加ボタン（+ボタン）をクリックし、機器名、機器の説明を入力する。
この際、機器 ID がすでに決まっているのであれば、それも入力する。
3. 機器 ID をメモする。（既に決まっているのであれば、特段シなくても良い）
4. Add ボタンを押す
5. 機器に対し、エージェントプログラムをインストールし、自動で起動するよう設定する。
その際、エージェントへの引数として、サーバーの IP アドレス、機器 ID を設定する。

- 機器を削除するとき

1. ブラウザからサービスにアクセスし、ログインする。
2. 画面から該当の機器をクリックし、削除ボタンをクリックする。
3. 「ほんとに削除するの？」というダイアログが出るので、OK を押し、画面から機器が削除されたことを確認する。

- 登録されている機器情報を変更するとき

1. ブラウザからサービスにアクセスし、ログインする。
2. 画面から該当の機器をクリックし、変更ボタンを押す。
3. 機器作成時と同じようなダイアログが表示されるので、機器名や機器情報を編集する。
4. OK ボタンをクリックし、機器の情報が変わったことを確認する。

- 機器の現在の状態を確認するとき

1. ブラウザからサービスにアクセスし、ログインする。
2. 該当の機器を確認する。
緑が稼働している状態、赤が稼働していない若しくは、稼働しているかわからない状態である。

- 機器の過去の状態を確認するとき

1. ブラウザからサービスにアクセスし、ログインする。
2. Log ページボタンを押す。
3. 機器のログ一覧が出るので、該当機器を探しだし、確認する。

第5章 検証と考察

作成したシステムを検証するために以下のような実験を行った。

5.1 実験概要

学内にて Wifi プローブパケットを利用した人流観測の実験を再現する

5.1.1 実験目的

作成したシステムが IoT サービスの開発にどう影響するのか観測する。

5.1.2 実験方法

第3者に Wifi プローブパケットを利用した人流観測のサービスを開発してもらい、その様子を観測する。また、観測後にインタビューを行い、システムを利用した場合としない場合の違いについて聞く。

5.2 準備

RaspberryPi, Wifi ドングル, プローブパケット観測ソフトウェア, 監視用エージェントを被観測者に提供する。被観測者は、それらを用いて、学内の各階に誰が居るのか、また、人の移動の様子を可視化するサービスを構築してもらう。

5.3 経過

5.4 考察

実験により、次のような評価を得ることができた。
評価から、有効であると分かった。

第6章 結論

本研究の背景には、IoT サービスにおける機器監視の煩わしさが挙げられる。具体的には、

1. IoT サービスに組み込む形で機器監視を行った場合、IoT サービス毎に監視部分を開発するので、コストが高い
2. 設置箇所が遠く、分散して設置されるため、定期的に確認しに行くことは現実的ではない

が挙げられる。

そこで、機器監視をサービスから独立させ、IoT 機器監視サービスとして提供すれば良いのではないかと考え、開発を行った。しかし、既存手法を調査する中で、IoT 機器は NAPT 環境下に置かれる事があることがわかり、IoT 機器から定期的に通知を送ることで解決を図った。

プロトタイプの開発の結果、下記のような知見を得ることが出来た。

- NAPT 環境下にある機器の状態を監視するために、機器から状態を定期的に通知する手法は有効であることが分かった。
- また、既存の機器監視サービスの多くは、機器の IP アドレスをサーバー側で管理しなければならないが、本手法を用いることで、IP アドレスに関係なく監視できることが分かった。

また、今後の課題としては、次のような事があることが分かった。

- ユーザビリティの向上
現状では、機器 ID やサーバー IP アドレスを IoT 機器に打ち込まなければならない、また、自動起動の設定も行わなければならない。
シェリスクリプトを実行すれば自動起動の設定まで行われる等、いっそうの簡略化を行う必要がある。
- 機器の識別問題の解決
現状では、どの機器が何だったのかまでは管理できていない。
QR コードを出力し、機器に貼り付けるなどによって、簡略化できると思われる。
- アラート機能の実装
アラートメールの送信などもできれば、常に監視していなくて良くなるのではないかとと思われる。

これらの課題を解決することで、IoT サービスの開発の手間を省けるのではないかと考える。

第7章 謝辞

第8章 参考文献

関連図書

- [1] 平成 27 年版 情報通信白書 (総務省) 「IoT の実現に向けたアプローチと我が国 ICT 産業の方向性」より <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc254150.html>
- [2] 6 割が「IoT は流行語」－エスキュービズム調査 (ZDBet Japan) <http://japan.zdnet.com/article/35093272/>
- [3] 先進テクノロジーのハイブ・サイクル 2011 年 Gartner <https://www.gartner.co.jp/press/html/pr20110907-01.html>