

# 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	IoT サービスの維持における問題	2
2.1	IoT サービスとは	2
2.2	IoT サービスの構造とは	2
2.3	IoT サービス維持の問題	3
第 3 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
3.1	IoT 機器の監視とは	4
3.2	IoT 機器の監視に必要な機能とは	4
3.3	IoT 機器の監視に求められる条件とは	4
3.4	IoT 機器監視に求められる要件とは	5
第 4 章	設計と実装	6
4.1	想定するユーザーの定義	6
4.2	課題と機能の関係	6
4.3	システムの構成	6
4.4	構成部品のそれぞれの動き	8
4.4.1	監視エージェントの動き	8
4.4.2	サービスの動き	8
4.5	ユーザーの動き？	8
第 5 章	検証と考察	10
5.1	実験概要	10
5.1.1	実験目的	10
5.1.2	実験方法	10
5.2	準備	10
5.3	経過	10
5.4	考察	10
第 6 章	結論	11
第 7 章	謝辞	12
第 8 章	参考文献	13

## 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 IoTサービスの維持における問題

### 2.1 IoTサービスとは

IoTとは、Internet of Thingsの略で「モノのインターネット」とも呼ばれる概念である。IoTでは、様々な物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、多様な自動化を行う。IoTサービスとは、ユーザーに対しIoTによる利便性を提供するものである。

IoTサービスは、半導体技術の進歩によりコンピュータが小型且つ安価になったこと、通信ネットワークの整備が進み様々な場所から安価に通信が利用可能になったことで登場した。

例えば、次のような物がある。

- コーヒーポット
- 太陽光発電の監視

コーヒーポットの自動化とは、コーヒーポットまで移動し、コーヒーポットの状態を確認しに行く手間を省くためのサービスである。このサービスの実現の為に、コーヒーポットに対してコンピュータを取り付ける。これらコンピュータが、コーヒー豆の残量やコーヒーが入ったか否かの情報をサーバーとやり取りする。それにより、サーバーはコーヒー豆の自動発注や、コーヒーが入ったことユーザーに知らせる。

太陽光発電の監視とは、太陽光発電所の発電量や機器の異常を確認しに行くための手間を省くためのサービスである。このサービスの実現の為に、太陽光発電所の機器にコンピュータを取り付ける。これらコンピュータが発電量や機器の異常の情報をサーバーとやり取りする。それにより、サーバーは、発電量や機器の異常をユーザーに知らせる。

このように、IoTサービスは、IoT機器とサーバーが連携し、ユーザーに利便性を提供するものである。今後も数多くサービスが登場すると考えられている。

### 2.2 IoTサービスの構造とは

IoTサービスは、IoT機器とサーバーが連携し利便性を提供するものである。IoTサービスの構造として、多数のIoT機器とサーバーがインターネットを介し連携する事が挙げられる。

IoT機器は、様々な環境へ設置され、周囲の状況を検知することや、周囲へなんらかの働きを行う為に使用される。コーヒーポットの例では、コーヒーが出来上がったか否かを検知する。また、コーヒーポットに対し、コーヒーを入れるよう働きかけている。この機器からの情報を収集し、処理しているのがサーバーである。サーバーは、IoT機器からの情報を蓄積・分析し、IoT機器やユーザーに対し何らかの働きかけを行う。コーヒーポットの例では、ユーザーがコーヒーを入れた回数からコーヒー豆の残量を分析し発注を行うよう働きかけている。サーバーにて動作するプログラムが、IoT機器と通信することで、IoTサービスを構成する。この通信に利用されるのがインターネットである。様々な通信リンクを用いてIoT機器とサーバー上のプログラムが連携する。

このように、IoTサービスの構造は、IoT機器とサーバー上のプログラムがインターネットを介し通信し、連携することで成り立っている。

## 2.3 IoT サービス維持の問題

IoT サービスは、IoT 機器とサーバー上のプログラムがインターネットを介し通信し合うことで成り立っている。IoT サービスを維持するためには、これらの構造を維持する必要がある。そのため、IoT 機器が正常に動作しているのか、通信が途切れていないか、監視する事が重要である。ところが、IoT 機器が多量に存在することや、IoT 機器が接続するネットワークが多様であることから、その監視には技術的困難がある。また、個別の IoT サービスに組み込まれた監視システム等を別として、一般的な監視サービスも存在しない。

IoT サービスでは、多量の IoT 機器を使用する。そのため、個々の IoT 機器を識別し、適切に管理することが困難である。また、家庭内や屋外等、様々な環境下に置かれる 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困難を極める。接続先のネットワークでもプライベートアドレスを付与されるなどの他、IoT 機器とサーバーとの通信に制限がかかる場合もある。そのため、遠隔から機器の状態を確認することが難しい。

このように、IoT サービスの維持において、IoT 機器の動作や通信状態を監視することは重要であるが、その監視には技術的な困難がある。IoT サービスの円滑な提供や今後の発展の為に、技術的な困難を越えた IoT 機器の監視の実現が重要である。

## 第3章 IoT機器からの通知に基づく機器監視サービスの提案

### 3.1 IoT機器の監視とは

IoT機器の監視とは、動作の状態・通信の状態を監視することである。動作状態とは、少なくとも電源が入っているのかどうか、( ? ) 通信状態とは、インターネットに接続され、サーバーと通信ができているのかどうかである。

このように、個々のIoT機器の動作状態、通信状態を確認することが、IoT機器の監視である。

### 3.2 IoT機器の監視に必要な機能とは

IoT機器の監視とは、個々のIoT機器の動作状態、通信状態を確認することである。そのため、次のような機能が必要となる。

- IoT機器の動作状態を一覧して表示する機能
- IoT機器の過去の状態を確認する機能

IoT機器の動作状態を一覧して表示する機能とは、現在のIoT機器の動作状態を見ることができる機能である。あるIoT機器が動作しなくなった時に、動作していないことを明確に示す必要がある。

IoT機器の過去の状態を確認する機能とは、過去のIoT機器の動作状態と通信状態を確認することができる機能である。過去の動作状態や通信状態を整理し、いつ動作していて、いつ動作していなかったのか、各IoT機器ごとに示す必要がある。

このように、IoT機器の監視には、IoT機器の動作状態を一覧して表示する機能、IoT機器の過去の状態を確認する機能を持つ必要がある。

### 3.3 IoT機器の監視に求められる条件とは

IoT機器の監視には、上記機能が必要である他に、IoT機器が設置されるネットワークにかかわらず監視ができることが求められる。何故ならば、IoT機器が設置されるネットワーク環境は多様であるからである。

IoT機器が設置されるネットワーク環境として、次のような環境が挙げられる。

- IoT機器に対しプライベートアドレスが与えられる環境
- IoT機器とサーバーの通信が制限される環境

多くの場合、IPアドレスの節約の観点から、IoT機器に対し、プライベートアドレスのみ与えられる。この場合、インターネット側からは、IPアドレスを指定することが出来ないため、通信は常にIoT機器から始まる必要がある。また、IoT機器とサーバーとの通信がセキュリティの観点から制限される場合もある。HTTP等一般的に広く使用される通信を除いてブロックされる事がある。

このように IoT 機器の監視には、IoT 機器から通信が始まることや、HTTP 等頻繁に利用される通信を利用しなければならないといった条件がある。

### 3.4 IoT 機器監視に求められる要件とは

これらを踏まえて、IoT 機器の監視には次のような機能要件が求められる。

- IoT 機器の動作状態を一覧して表示する機能
- IoT 機器の過去の状態を確認する機能

また、非機能要件として、次のような制約がある。

- プライベートアドレスが与えられたとしても動作する事
- インターネットの通信として一般に広く利用されている HTTP 等の通信を用いる事



## 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  
シェルスクリプトーツ ( 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考えられる。

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>