

# 修士論文

## 題目

IoT サービスにおける IoT デバイス監視簡単化サービスの提案

## 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	序論	1
1.1	研究の背景	1
1.2	問題	1
1.3	研究の目的	2
第 2 章	既存の解決策とその課題	3
2.1	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	3
2.1.1	デバイスにディスプレイとキーボードをつける	3
2.1.2	PC を持参し、接続する場合	3
2.1.3	遠隔からログインし、状態を確認する場合	3
2.1.4	デバイスに LED 等の簡素なディスプレイをつける	4
2.2	設置後、正常に稼働しているのか確認するのが面倒	4
2.2.1	デバイスを直接現地に行って確認する	4
2.2.2	遠隔から確認する	4
2.3	いつ稼働していて稼働していなかったのか管理するのが大変	4
2.3.1	ログファイルを回収して確認	4
2.3.2	Fluentd Elasticsearch Kibana を使う	4
2.3.3	Telegraf + Influxdb + Grafana を使う	5
第 3 章	提案する解決策	6
第 4 章	提案システムの仕様と構成	7
第 5 章	提案システムの実装	8
第 6 章	検証と評価	9
第 7 章	考察	10
第 8 章	結論	11
第 9 章	謝辞	12
第 10 章	参考文献	13

## 概 要

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・セキュリティーの問題・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 序論

本章では、研究の背景及び現状の課題について記述し、本研究の目的について述べる。

## 1.1 研究の背景

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、家庭へのインターネットの普及により、全ての物がインターネットに接続し相互に情報を交換し合い様々な自動化を実現する IoT が注目されている。

このように、IoT サービスの開発が盛んに行われている。

## 1.2 問題

このように、IoT デバイスの価格が下がることで、IoT サービスの開発にかかるコストが低減され、開発への垣根が下がる一方で、サービスの運用において、次のような問題がある。

- 数が多くて管理しきれない問題
  - － 設置前の設定において、どのデバイスをどこに設置すれば良いのかわからなくなる -i ラベリングにて解決
  - － 設置後、どのデバイスがどこに設置されたのかわからなくなる -i 帳簿をつけることで解決
  - － 設定の際に、個別の設定をしなければならないのが面倒  
具体的には、デバイスに振る ID 等。  
ラベリングと整合性が取れていなければならない。
- 稼働状況の監視が面倒な問題
  - － 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、デバイス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。

稼働状況の監視については、IoT サービスで行うことがある程度可能だが、サービス自体に手を加える必要があるため、開発のコストが高くなる。

その中で、私は、IoT デバイスの状態監視に着目した。このような解決策がとられているが、問題が多い

### 1.3 研究の目的

そこで、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、これらの問題を解決できるのではないかと考えた。本研究では、IoT デバイスの監視サービスを開発することで、IoT デバイスの状態監視を簡単化することを目的とする。

## 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、デバイス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。

それぞれにおける既存の解決策と利点と欠点を以下に述べる

### 2.1 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

#### 2.1.1 デバイスにディスプレイとキーボードをつける

個々のデバイスごとにつける場合

デバイス1台あたりのコストがかさむ勝手に操作されることがある（セキュリティの問題）

設置者が持参しつける場合

デバイスにディスプレイを接続できる必要がある。より安価なデバイスだと、ディスプレイを接続できない場合が多い。勝手にディスプレイ等をつけ、操作されることがある（セキュリティの問題）

#### 2.1.2 PC を持参し、接続する場合

PC と接続できるインターフェースを用意する必要がある。勝手に PC を繋がれるかもしれない。  
上記2つは勝手に操作される可能性がある。

#### 2.1.3 遠隔からログインし、状態を確認する場合

間に NAPT が挟まっている場合があるのでできない事がある。

また、上記3つとも共通だが、デバイスに対して知識のある人物が確認しなければならない。設置場所が離れている場合は、それ相応のコストがかかる。

### 2.1.4 デバイスに LED 等の簡素なディスプレイをつける

最も安価だが、デバイスに LED をつけるのが、（そこまででも無いけど）めんどい。また、詳しいことはわからない。

## 2.2 設置後、正常に稼働しているのか確認するのが面倒

### 2.2.1 デバイスを直接現地に行って確認する

設置場所が離れている場合が多いので、コストがかかる。

### 2.2.2 遠隔から確認する

ICMP Ping を用いる

NAPT が障害となり、遠隔から確認することは無理。

SNMP を用いる

NAPT が障害となり、遠隔から確認することは無理。

Fluentd Elasticsearch Kibana を使う。

Fluentd が入るようなデバイスでないといけない。また、サービス毎にグラフ描画の設定をしなければならない。

Zabbix

（調査中？）

Telegraf + Influxdb + Grafana

Telegraf が入るようなデバイスでないといけない。また、サービス毎にグラフ描画の設定をしなければならない。

## 2.3 いつ稼働していて稼働していなかったのか管理するのが大変

### 2.3.1 ログファイルを回収して確認

現地に行くのめんどい

### 2.3.2 Fluentd Elasticsearch Kibana を使う

Fluentd が入るようなデバイスでないといけない。また、サービス毎にグラフ描画の設定をしなければならない。

### 2.3.3 Telegraf + Influxdb + Grafana を使う

Telegraf が入るようなデバイスでないといけない。また、サービス毎にグラフ描画の設定をしなければならない。



## 第3章 提案する解決策

## 第4章 提案システムの仕様と構成

## 第5章 提案システムの実装

## 第6章 検証と評価

## 第7章 考察

## 第8章 結論

## 第9章 謝辞

## 第10章 参考文献