

# 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	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	デバイス設置箇所に行って、直接確認する	3
2.2	ICMP Ping を活用する	3
2.3	SNMP を利用する	4
2.4	Zabbix を使用する	4
2.5	Fluentd Elasticsearch Kibana を利用する	4
2.6	Telegraf Influxdb Grafana 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
3.2	学内での wifi を用いた人流観測	5
3.3	要件の抽出	5
第 4 章	設計と実装	6
4.1	システムの構成	6
4.2	想定するユーザーの定義？	6
4.3	ユーザーの動き？	6
第 5 章	検証と考察	7
5.1	実験概要	7
5.1.1	実験目的	7
5.1.2	実験方法	7
5.2	準備	7
5.3	経過	7
5.4	考察	7
第 6 章	結論	8
第 7 章	謝辞	9
第 8 章	参考文献	10

## 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。

ここで、IoT サービスとは、IoT による自動化を提供する物のうち、デバイスから得た情報を蓄積・分析し、結果を元に、表示等の動作を行うものと定義する。IoT サービスの構成としては、複数のデバイスから1つのコンピューターへ情報を送り、その上で蓄積・解析し、結果を表示する等の動作を行っているものが多い。

## 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。

### 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。

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

### 2.1 デバイス設置箇所に行って、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、デバイス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。

### 2.2 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 デバイス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 デバイスの IP アドレスは、IoT デバイス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。

## 2.3 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の時間、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。

## 2.4 Zabbix を使用する

## 2.5 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、それぞれに対して設定が必要になる。また、Fluentd が動くデバイスである必要がある。グラフの作成が手動でめんどくさい。。

## 2.6 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してくれます。Telegraf とは Influxdb へ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す。Grafana とは可視化ツールの一つで、グラフの描画等が行えます。これも、FluentdElasticSearchKibana の時のように、それぞれに対して設定が必要で、デバイスの上で Telegraf が動く必要が有ります。また、グラフの描画も手動でめんどくさいです。

## 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。そのために、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。RaspberryPi と ampsence を用いて、その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してもらい、インターネットとの接続は、商店に敷設済みの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。

#### 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、分析する段階に置いて、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。
- 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、遠隔からアクセスし、稼働状態の確認を行った。
- 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。

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。

### 3.3 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 デバイスの稼働状態がわかる  
IoT デバイスの稼働状態は、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。
- IoT デバイス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。

そこで、IoT サービスとは独立した、IoT デバイスの稼働状態を監視・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。

### 4.1 システムの構成

システムの構成は以下のとおりである。

### 4.2 想定するユーザーの定義？

### 4.3 ユーザーの動き？

ユーザーの動きは以下のとおりである。

## 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

#### 5.1.1 実験目的

#### 5.1.2 実験方法

### 5.2 準備

### 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

よって、このアプローチは有効であるということが分かった。  
今後の課題としては、次のような物があることが分かった。

- 仮

## 第7章 謝辞

## 第8章 参考文献