

インターネット上の Wiki ページにより NAT 背後のセンサ端末の設定変更や制御が可能な IoT システムによるサーバとサーバ室の監視

山之上 卓[†]

概要: インターネット上の Wiki ページにより, NAT (Network Address Translator router) 背後のセンサ端末の設定変更や制御が可能な IoT (Internet of Things) システムを利用して, サーバとサーバ室の監視を行うことについて述べる. ここで利用する IoT システムはインターネット上の Wiki ページと, Android 端末や Raspberry Pi に各種センサを接続して作られた Bot (Wiki Bot) で構成されている. Wiki Bot は NAT 背後に設置可能であり, このことにより, インターネットから行われる様々な攻撃に対する防御能力を大きくすることができる. また, 外部からは直接アクセスできない, NAT 背後にある, サーバの状況を Wiki Bot によって把握し, この状況をインターネット上で閲覧可能にできる. Wiki Bot は Wiki ページ上に書かれたコマンド列とプログラムで構成されたスクリプトを定期的に読み込む. そのスクリプトによって, Wiki Bot に接続されたセンサの設定を行い, データを入力し, 簡単な処理を行い, 処理されたデータを Wiki ページに書きこむ. スクリプトに従って Twitter でつぶやくことも可能である. Wiki Bot は管理者が Bot のプログラムを, Bot のある場所に行って, 直接操作することなしに, インターネット上にある Wiki ページのスクリプトを書き換えることにより, Wiki Bot のセンサの設定変更や, データのサンプリング間隔などの調整を行うことができる.

キーワード: Wiki, Bot, IoT, 監視

Monitoring a Server and a Server Room by an IoT System, which Can Configure and Control its Sensor Terminals Behind NAT using Wiki Page on the Internet

Takashi YAMANOUE[†]

Abstract: This paper discusses a method of monitoring servers or server rooms by an IoT (Internet of Things) system which enables configure and control of a sensor terminal behind a NAT (Network Address Translator router) by a Wiki page on the Internet. This IoT system consists of the Wiki pages on the Internet and a Bot (Wiki Bot) which is made by combining an Android terminal or a Raspberry Pi, and sensors. The Wiki Bot can be placed behind the NAT. The Wiki Bot has higher tolerance of various attacks from the Internet by placing the Wiki Bot behind the NAT. The IoT system also can monitor servers, which is placed behind the NAT, from the Internet. The bot is controlled by the script, which is a sequence of commands and the program, on the Wiki page. The Wiki Bot acquires data from sensors which are connected to the Wiki Bot, and processes the data by the sequence of commands and the program. Settings of sensors and the sampling rate of data can be changed by rewriting the script on the Wiki page, without the manager going to the place of the Wiki Bot and rewriting the settings at the place.

Keywords: Wiki, Bot, IoT, Monitoring

はじめに

ICT 基盤の管理者は, 常時, 多くのサーバやサーバ室を監視している. サーバやサーバ室の監視は, ICT 基盤管理の未経験者にとっては, 簡単な仕事に見える. しかしながら, これは重要で大変な仕事である. サーバやサーバ室の潜在的な障害の発見に失敗し, その結果, 障害への対応に失敗すると, 利用者は仕事ができなくなる. また, 建物の火災など, 深刻な問題が発生する可能性もある.

人間は完全ではない. 従って ICT の管理者はサーバ

やサーバ室の障害の発見を失敗してしまう可能性がある. このような失敗を少なくするため, 管理者は様々な手法を利用しているが, その手法の一つとして, インターネット上の Wiki ページにより, NAT (Network Address Translator router) 背後のセンサ端末の設定変更や制御が可能な IoT (Internet of Things) システムを利用することについて述べる.

ここで使用する IoT システム (Wiki IoT) はインターネット上の Wiki ページと, Raspberry Pi や Android 端末等に各種センサを接続して作られた Bot (Wiki Bot) で構成されている. Wiki Bot は NAT 背後に設置可能で

[†] 福山大学
Fukuyama University

あり、このことにより、インターネットから行われる様々な攻撃に対する防御能力を大きくすることができる。また、NAT 背後にあるサーバの状況をこの Bot によって把握し、この状況をインターネット上で閲覧可能にできる[1][2][3]。Wiki ページを IoT システムのデータベースとして利用することにより、機械(Wiki Bot)も人間も同じ書式のデータを読み書きすることができる。Wiki ソフトウェアとして PukiWiki[4]を利用しているため、データベースシステムを別途用意することなしに、IoT システムのデータ蓄積部を構築することができる。

Wiki Bot は Wiki ページ上に書かれたコマンド列とプログラムで構成されたスクリプトを定期的に取り込み、そのスクリプトによって、Wiki Bot に接続されたセンサの設定を行い、データを入力し、簡単な処理を行い、処理されたデータを Wiki ページに書きこむ。スクリプトに従って Twitter でつぶやくことも可能である[5]。

Wiki Bot は Bot のプログラムを、管理者が Bot のある場所に行き、直接入れ替えることなしに、インターネット上にある Wiki ページのスクリプトを書き換えることにより、NAT 背後にある Wiki Bot のセンサの設定変更や、データのサンプリング間隔などの調整を行うことができる[6]。

我々は、Wiki IoT を使って、サーバの監視とサーバ室の監視を行っている。Wiki IoT で使っている Wiki Bot として、Raspberry Pi[7]に I2C センサを接続し、その I2C センサの設定を Wiki ページ上のスクリプトで行うことができるものもある[8]。

本論文では、Wiki IoT と Wiki Bot の概要、Wiki IoT を使ったサーバ監視の例、Raspberry Pi に I2C センサ等を接続して構成した Wiki Bot によって、サーバ室の監視を行っている例、関連研究について述べる。

2. Wiki IoT と Wiki Bot の概要

図 1 に、Wiki IoT の概要を示す。Wiki IoT は、インターネット上の Wiki ページと、センサが接続された Wiki Bot や、センサネットワークと通信が可能な Wiki Bot/Gateaway[9] や、センサやセンサネットワークが接続されていない Wiki Bot から構成されている。センサだけでなく、アクチュエータを使うこともできる。

図 2 に、Wiki Bot の概要を示す。Wiki Bot はスクリプトインタプリタであり、Wiki Bot の振る舞いは、Wiki ページに記述されたスクリプトによって定まる。Wiki ページのスクリプトは、コマンドの列とプログラムで構成される。

Wiki Bot のハードウェアとして、Android 端末[10]にセンサやアクチュエータを付けた Arduino ボード

[11]を接続したもの[1][2]、Raspberry Pi の GPIO にセンサやアクチュエータを接続したもの、GPIO には何も接続していない Raspberry Pi、Windows パソコン、などが利用できる。

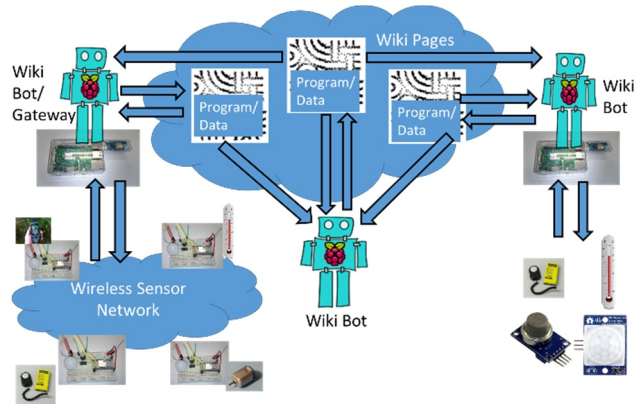


図 1. Wiki IoT の概要

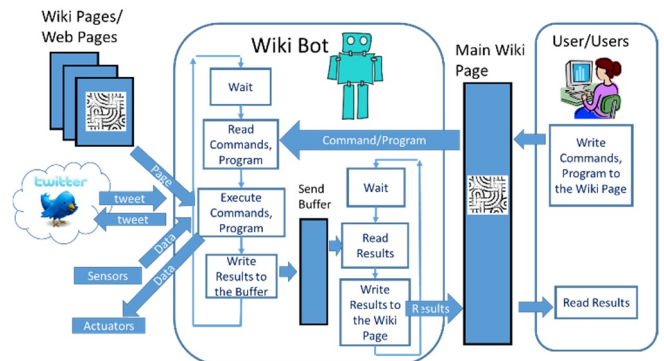


図 2. Wikii Bot の概要

```
objectPage http://www.
device yamaRasPiDp9_1 or yamaRasPiDp9_2 start after no w
command: set readInterval=60000
command: set execInterval=0
command: clear sendBuffer;
command: program ex1
program: s=0;
program: for i=0 to 10
program:   s=s+i
program:   ex("service","putSendBuffer "+s)
program: next i
command: end ex1
command: run ex1
command: ex("service","sendResults.")
result:
0
1
3
6
10
15
21
28
36
45
55
currentDevice="yamaRasPiDp9_1",Date=2018/5/15/ 12:54:17
```

図 3. Wiki Bot のプログラムと実行例

Wiki Bot は NIC や Wi-Fi を通じてインターネット上の Wiki ソフトウェアと通信を行う。実行時には、Wiki

Botに接続されたセンサや、Wiki Botに接続された無線センサネットワーク(WSN)を制御し、データを入力するが、センサや無線センサネットワークを持たないWiki Botも存在する。このWiki Botは、Wiki IoTのWikiページや、その他のページに書かれたデータを解析するために利用される。

Wiki Bot側から定期的にそのスクリプトが記述されたWikiページを読みに行くため、多くの場合は、Wiki BotとWikiページの間にはNATやファイヤーウォールが存在しても、Wikiページに書かれたスクリプトによって、Wiki Botを制御することができる。Wiki BotがWikiページを読みに行く間隔も、スクリプトで変更することができる。

図2は、Wiki Botの振る舞いも示している。Wiki Botが起動した後、Wiki Botは以下の手順を繰り返す。

1. 指定されたWikiページに書かれたスクリプトを読み込む。このWikiページのことを、Objectページと呼ぶ。ObjectページのURLはWiki Botの設定ファイルに書かれている。Wiki BotのGUIでこのURLを設定することもできる。

2. スクリプトを実行する。このとき、取り付けられたセンサやWSNの先にあるセンサの設定を行ったり、センサからデータを入力したり、Wikiページや他のWWWのページのデータを入力したり、入力したデータを処理したり、Twitterにアクセスしたりする。

3. スクリプトを実行後、実行結果を指定されたObjectページに書きこむ。

図3にWikiページに書きこまれたスクリプトとその実行結果の例を示す。このスクリプトはWiki Botに対して以下のような指示を行う。:

- ・ “command: set readInterval=60000”の行により、1分間に一度、このObjectページをWiki Botが読み込むようにする。

- ・ 以下部分でプログラムを定義する。

- ・ command: program ex1

- ・ program: ...

- ・ ...

- ・ program: ...

- ・ command: end ex1

- ・ “command: program ex1”と“command: end ex1”の間が定義されるプログラムで、ex1と名付けられてWiki Botに格納されることを示す。“program:”で始まる行が、その行がプログラムの一部であることを表す。このプログラムは1から10までの和を、途中経過を表示しながら計算するものである。

- ・ “command: run ex1”によって、格納されたプログラムex1が実行される。

このプログラムは、内部ではLispに変換して動作

させているため再帰的な関数の定義と実行が可能である。配列としてハッシュ表を利用できる。表計算に関する関数もいくつか利用可能であり、CSVを表を読み込んで、データ解析や加工を行うことが可能である。

Objectページに書かれたデータの量が大きくなると、利用者がこれを読む負荷が大きくなる。同時に、Wiki Botもこれを読んだり、データを書き加えたりする負荷が大きくなる。この問題への対処方法として、ページの分割を行うことができる。指定した時間に達したり、Objectページのデータが指定した長さに達したりしたら、コマンド“set pageName=<next page name>”を実行することにより、次にWiki Botが読み込むスクリプトやデータが書きこまれたObjectページを、<next page name>に変更可能である。<next page name>の中に、<hour>や<day>が書かれていた場合、<hour>はその時の時間の時に変換され、<day>はその時の日に変換される。あらかじめ、これらのページを用意しておくことで、時間ごとや日ごとにデータを分類することができる。

複数のObjectページで構成されたWiki IoTの場合、いくつかのObjectページ内に、同じスクリプトを持つ場合がある。同じ記述を多くのObjectページで行うのは無駄があるし、そのスクリプトを変更するとき、すべてのObjectページの同じ部分を変更するのは大変であるので、オブジェクト指向プログラミング言語の場合と同様に、同じ記述の部分を一か所のWikiページにまとめ、ObjectページにそのWikiページを挿入することができる。挿入されるページのことをClassページと呼ぶことにする。スクリプトの、挿入したい行に、ClassページのURLを指定する“include”コマンドを書くことにより、ObjectページにClassページを挿入することができる。

Wiki IoTは、Cross Over includeとCross Over executionを導入することにより、高可用性を持たせることもできる。2つの同じClassページをPrimary ClassページとSecondary Classページとして異なるサーバに置いておき、Primary ClassページのWikiソフトウェアに障害が発生した場合、Objectページは、次にClassページを読み込むとき、Secondary Classページの内容を読み込む。Objectページについても、Primary ObjectページとSecondary Objectページを用意しておき、Primary Objectページに障害が発生した場合、Wiki Botは次にSecondary Objectページを利用できる。このとき、計算途中のデータは、新しいページに自動的に引き継がれる。Wiki Botについても、Primary Wiki BotとSecondary Wiki Botを利用することにより、片方のWiki Botに障害が発生した場合、障害が発生していない側のWiki Botが代わりに動作を開始する(図4)。

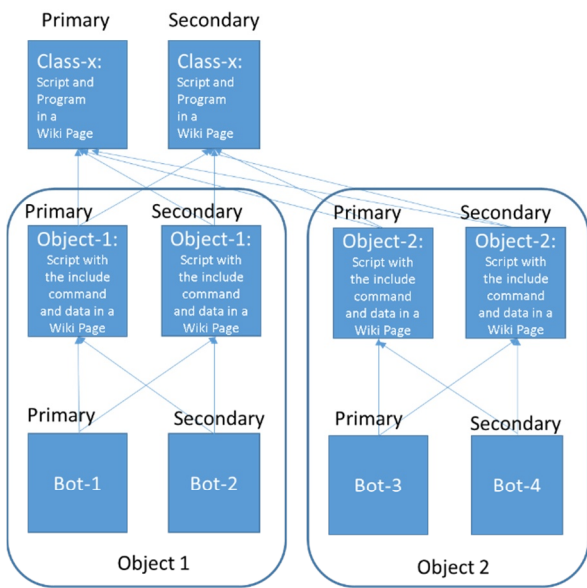


図 4. Cross Over include と Cross Over execution

3. 外部から直接アクセスできないサーバの監視

組織のコンピュータネットワークでは、セキュリティ強化や、それに導入されているライセンスの制限等のため、NAT やファイヤーウォールにより、外部からのアクセスを禁止しているサーバを持っている場合がある。このようなサーバは、多くの場合、外部から直接、その稼働状況等を確認することができない。New relic[12] のように、監視プログラムを目的サーバにイ

nstallすることで、クラウド上のサーバで目的サーバの稼働状況を確認する方法もあるが、ハウジングサービスを利用している顧客サーバのように、管理者側の都合によるソフトウェアのインストールが難しい場合もある。本サーバ監視システムで死活監視を行う場合は、Wiki Bot を、監視対象と同じ NAT セグメント内に設置するだけで良く、目的サーバにプログラムをインストールしたり設定変更したりする必要はないため、そのような問題も解決できる。

福山大学の一部の授業は、情報倫理デジタルビデオ小品集[13]を使って行われている。このビデオを学内のサーバに格納して使っているが、このサーバは、ライセンスの制限により、それにアクセスできるクライアントの IP アドレスの制限を行っており、また、セキュリティの強化を行う為、学外からはアクセスできないようにしている。

このサーバの死活監視を行うために、Wiki IoT を使っている[14]。Wiki Bot を学内 LAN に設置し、この Wiki Bot の IP アドレスを、監視対象のサーバにアクセス可能なクライアントとして登録している。学外から閲覧可能なサーバに、この Wiki Bot の Object ページを用意し、Wiki Bot に動作指示を行うと同時に、Wiki Bot 自体の稼働状況も学外から把握できるようにしておく。この Object ページは、監視対象のサーバの生死を、定期的に Twitter で tweet するスクリプトを持っており、管理者は、Twitter の Tweet の状況を確認することで、学外から監視対象のサーバの生死を確認することができる(図 5)。

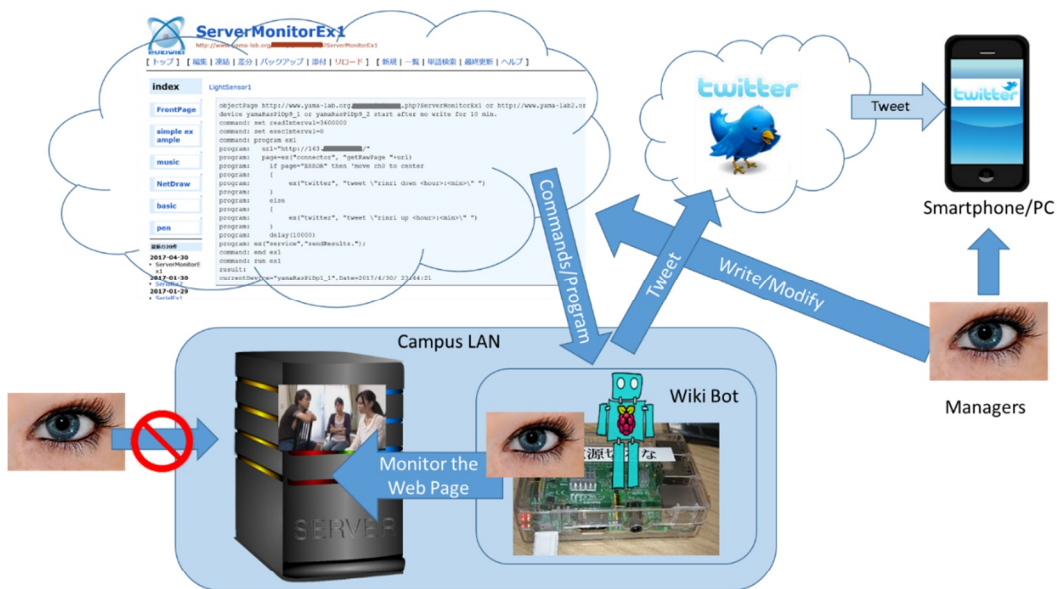


図 5. 外部から直接アクセスできないサーバの監視システム



ServerMonitorEx1

<http://www.yama-lab.org/...?ServerMonitorEx1>

[トップ] [編集 | 凍結 | 差分 | バックアップ | 添付 | リロード] [新規 | 一覧 | 単語検索 | 最終更新 | ヘルプ]

index

LightSensor1

FrontPage

simple ex
ample

music

NetDraw

basic

pen

最新の20件

- 2017-04-30
 - ServerMonitorEx1
- 2017-01-30
 - SerialEx2
- 2017-01-29
 - SerialFx1

```
objectPage http://www.yama-lab.org/...php?ServerMonitorEx1 or http://www.yama-lab2.or
device yamaRasPiDp9_1 or yamaRasPiDp9_2 start after no write for 10 min.
command: set readInterval=3600000
command: set execInterval=0
command: program ex1
program: url="http://163. ..../"
program: page=ex("connector", "getRawPage "+url)
program: if page="ERROR" then 'move ch0 to center
program: {
program:     ex("twitter", "tweet \"rinri down <hour>:<min>\" ")
program: }
program: else
program: {
program:     ex("twitter", "tweet \"rinri up <hour>:<min>\" ")
program: }
program: delay(10000)
program: ex("service","sendResults.");
command: end ex1
command: run ex1
result:
currentDevice="yamaRasPiDp1_1",Date=2017/4/30/ 23:44:21
```

図 6. 外部から直接アクセスできないサーバの監視システムを制御する Wiki ページ

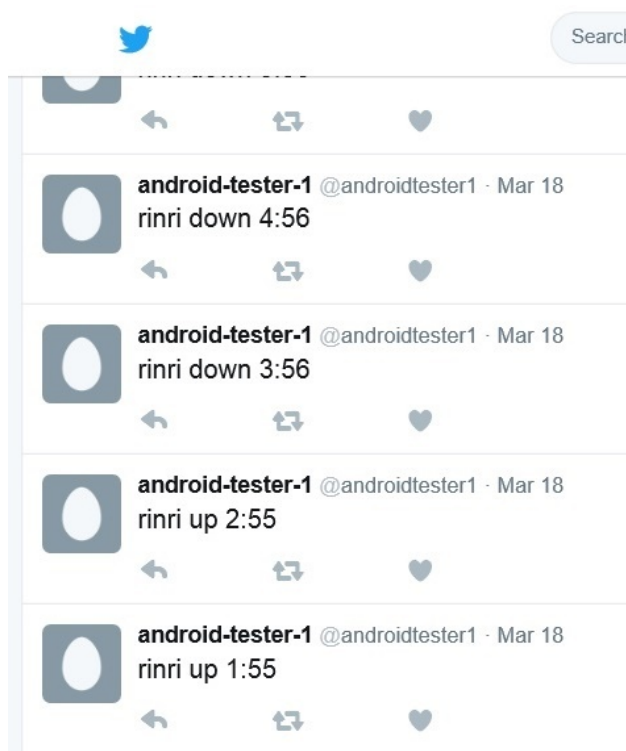


図 7. サーバの生死のつぶやき

図 6 に、サーバの生死を確認し、その状況を Twitter でつぶやくスクリプトが記述された Wiki ページを示す。図 7 は、3 月 18 日にサーバがダウンしたことを Twitter の Tweet で通知した場合を示す。

4. サーバ室の監視

我々は IOT23 において、「Wiki で設定やプログラムを記述可能なセンサネットワークシステム」について発表を行っている[6]が、このシステムのセンサ部分をサーバ室に置くだけで、サーバ室の温度などの状況を監視し、外部からそれを確認することができる。

我々は、文献[6]の Wiki IoT を使って、福山大学工学部のサーバ室の監視を行っている。このサーバ室ではおよそ 10 台のファイルサーバや AD サーバなどが稼働している。以前、週末に空調設備の障害が発生し、月曜日の朝の授業のとき、温度異常のためサーバ室内の各種サーバに障害が発生していて、授業がおこなえなかったことがあった。幸い、監視を開始してから、空調の異常は発生していないが、いちいち鍵を開けて入らないと把握できなかったサーバ室の状況を、学外も含めた外部から常に監視可能になった。ここで使っている Wiki ページはベーシック認証により、閲覧制限を行っている。

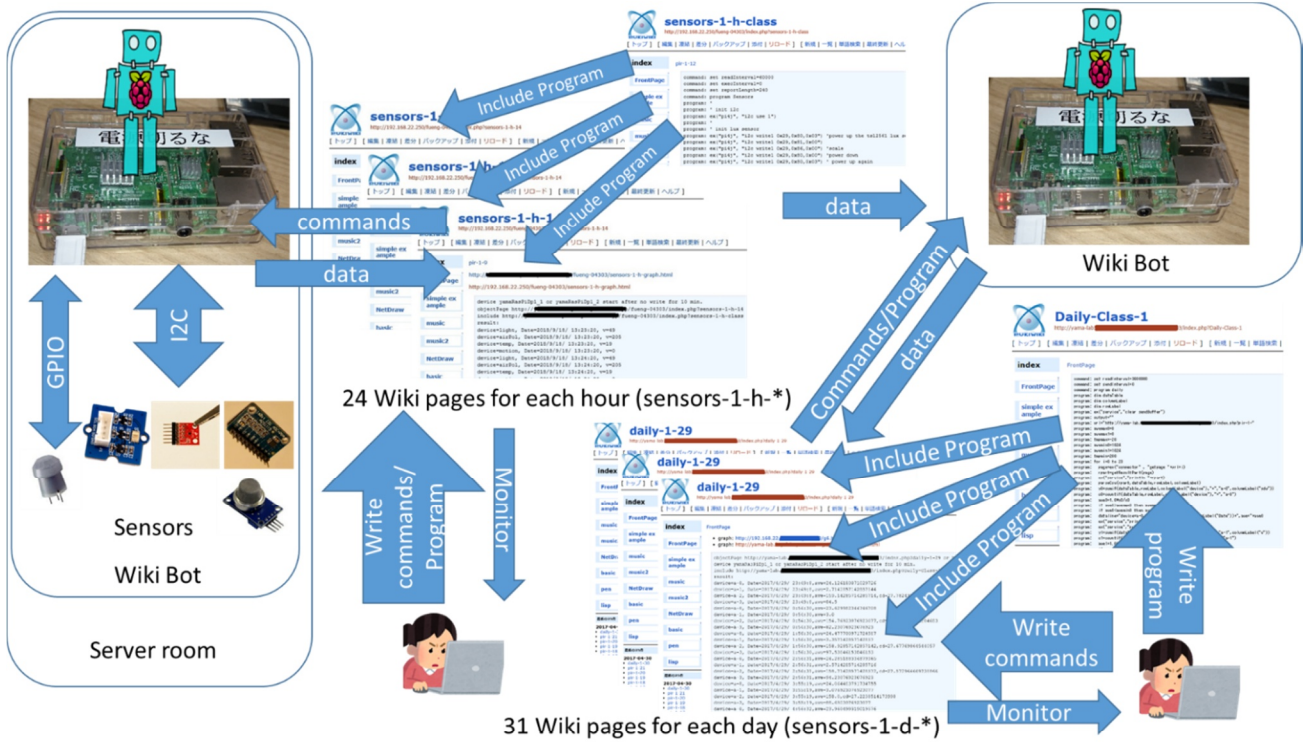


図 8.サーバ室の状況を監視する IoT システムの概要

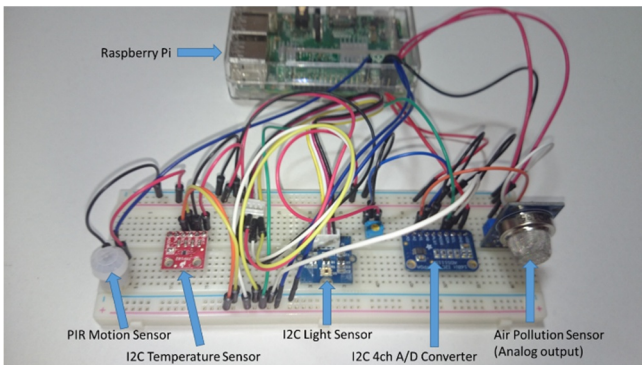


図 9. 温度、照度、空気質、人の動きを検知する Wiki Bot

<http://192.168.22.250/fueng-04303/sensors-1-h-graph.html>

```
device yamaRasPiDp1_1 or yamaRasPiDp1_2 start after:
objectPage http://[redacted]/fueng-04303/
include http://[redacted]/fueng-04303/
result:
device=light, Date=2018/9/18/ 14:1:21, v=49
device=airPol, Date=2018/9/18/ 14:1:21, v=205
device=temp, Date=2018/9/18/ 14:1:21, v=19
device=motion, Date=2018/9/18/ 14:1:21, v=0
device=light, Date=2018/9/18/ 14:2:22, v=49
device=airPol, Date=2018/9/18/ 14:2:22, v=205
```

図 10. Object ページ “sensors-1-h-14”の一部

```
program: 'get the Analog value from the ADS1115 ch1
program: config=0x0003|0x0000|0x0000|0x0000|0x0080|0x0100
program: config=config|0x0000
program: config= config|0x5000 'channel 1
program: config=config|0x8000
program: ex("pi4j", "i2c write2 0x49,0x01,"+config)
program: delay(10)
program: v2=ex("pi4j", "i2c read2 0x49,0x00")
program: airpol=s2i(v2)
program: airpol=airpol+800
program: '
program: ' get temperature
program: v1=ex("pi4j", "i2c read1 0x48,0x00")
program: tmp=s2i(v1)
program: .
```

図 11. 図 8 のセンサからのデータ入力を行う Wiki IoT のスクリプトの一部

当初、この Wiki Iot は、IOT23 のシステムにおいて、Android 端末で実現されていたセンサデータ収集部分の Wiki Bot をそのまま利用していた[14]が、今回、Wiki Bot を、Android 端末から Raspberry Pi で作った I2C インターフェースが利用可能な Wiki Bot[8]に置き換えたものを作成し、置き換えた。図 8 にこの Wiki IoT の概要、図 9 に Raspberry Pi で作成した Wiki Bot のハードウェアの写真を示す。

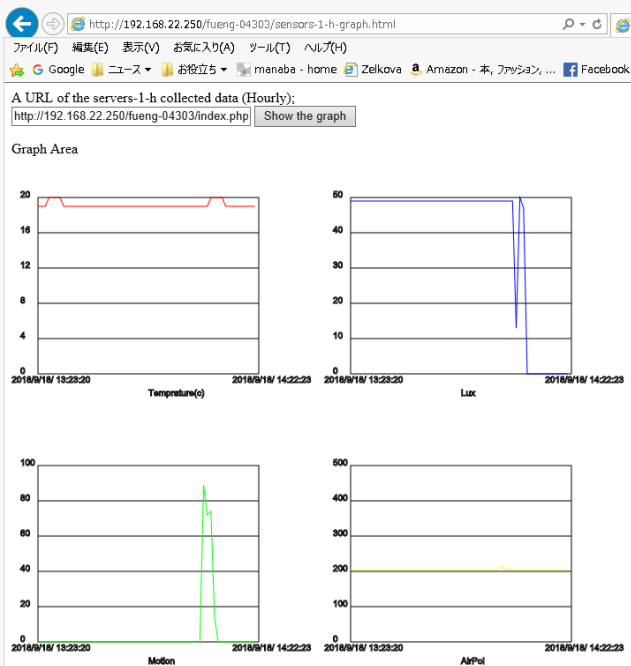


図 12.: Wiki IoT で収集されたサーバ室の温度・照度・人の動き、空気の汚れの推移の例。

図 8 において、サーバ室に設置された Wiki Bot の I2C インターフェースに、照度センサ、温度計、AD 変換素子が接続されている。AD 変換素子の先には空気汚染度センサが接続されている。人間の動きを検知する PIR センサが Raspberry Pi の GPIO 端子に接続されている。この、サーバ室の Wiki Bot は、時間 x に対応した Object ページ “sensors-1-h-x” を定期的読み込む(図 10)。この Object ページの中で、Class ページ “sensors-1-h-class” を include している。この Class ページの中に、センサの設定やデータ入力を行うプログラムが記述されている。このプログラムに従ってデータ入力した Wiki Bot は、その結果を、Object ページ “sensors-1-h-x” に書き加える。この Class ページには、時間ごとに、読み込む Object ページを指定するコマンドが記述されているので、それぞれの時間に対応したページに、その時間の間のデータが書きこまれることになる。

図 8 の右側の Wiki ボットは、日 y に対応した Object ページ “Daily-1-y” を定期的読み込む。この Object ページの中で、Class ページ “Daily-Class-1” を include している。この Class ページは、”sensors-1-h-y” のそれぞれの時間の、それぞれのセンサの値の平均を取って、Daily-1-y に書き加える。この Class ページには、一日ごとに、その日に対応した Object ページを指定するコマンドが記述されているので、それぞれの日付に対応したページに、その日の 24 時間のデータの推移が書きこまれることになる。

図 11 に、図 10 の Object ページに挿入される”sensors-1-

h-class” ページの一部を示す。この部分は、I2C インターフェースを持つ AD 変換素子に対して、感度やサンプリング間隔やデータチャンネルの設定を行い、空気汚染度センサのアナログ出力が接続されたチャンネルからデータを入力し、そのデータ値を補正することと、I2C インターフェースを持つ温度計から温度データを入力する部分を示している。

図 12 に、sensors-1-h-14 に書きこまれた、Wiki Bot が獲得した温度、照度、人の動き、空気汚染度の推移の例を示す。図 10 で示す Object ページの一番上の行に、そのページのデータをグラフ表示する JavaScript ページへのリンクがあり、この部分をクリックすることで、図 11 のグラフが表示される。

5. 関連研究

5.1 基幹ネットワークの設定による監視

外部から NAT 内の機器の死活監視を行う手段として、基盤ネットワークに監視用セグメントを設け、監視するための機器に 2 つの NIC を持たせ、片方を NAT 内の LAN に接続して監視対象を監視し、もう片方を監視用セグメントに接続し、外部からの死活を確認可能にする方法が考えられる。既存の基盤ネットワークに新たに監視セグメントを作る場合や監視セグメントの拡張を伴う場合、組織間調整、基盤ネットワークの設計、情報コンセントの新規設置工事、設定変更などが伴い、気軽には実行できない。VLAN を使って同様の監視を行うこともできるが、このときは監視対象サーバの利用者のみで実施可能である。しかしながら、VLAN の設定が必要になる。これに対して我々の Wiki IoT は、基盤ネットワークの設定変更も VLAN の設定も必要ない。

5.2 Kaseya と UNIFAS

パソコンの遠隔管理システムの KASEYA[16] と Wi-Fi アクセスポイントの管理システムの UNIFAS[17] は、パソコンや Wi-Fi アクセスポイントに実装されたエージェントプログラムと、パソコンや Wi-Fi アクセスポイントを管理する Web サイトによって構成されている。この構成は我々の Wiki IoT と類似しており、Kaseya も UNIFAS も Wiki IoT も、NAT 背後のエージェントや Wiki Bot を Web ページで制御可能である。しかしながら、Kaseya や UNIFAS が独自の Web サーバを必要とするのに対して、Wiki IoT の Wiki ページはオープンな PukiWiki を利用している。このため、我々の Wiki IoT は特定の企業の製品に依存しないシステムを構築することが可能である。

5.3 Obniz

Obniz[18] は、モーターやセンサーなどのつながれた電子部品がブラウザからプログラムできるようになる開発ボードである。Obniz は Wi-Fi を利用して、Obniz 独自の Cloud (Obniz Cloud) に接続される。Obniz のプログラムは HTML と

して保存され公開される。ユーザーがそれをブラウザで開くことで、ブラウザ内で実行され、遠隔で obniz や、obniz につながれた部品が操作される。Obniz も、Kaseya や UNIFAS の場合と同様に、NAT 背後の機器を、Web から遠隔操作可能である。また、我々の Wiki IoT と同様に、I2C デバイスの細かな制御を行うことが可能である。

Obniz は WebSocket を使った Obniz-Obnize クラウド間の通信を行うことが可能なのに対して、我々の Wiki IoT は WebSocket の通信機能は持っていない。Obniz は JavaScript をプログラミング言語として利用できるのに対して、われわれの Wiki IoT は、独自のプログラミング言語を使っている。Obniz も Kaseya や UNIFAS と同様に独自の Web サーバ (Obniz Cloud) を必要とするのに対して、Wiki IoT の Wiki ページはオープンな PukiWiki を利用している。Obniz を制御するプログラムは、Obniz ではなく、ブラウザやサーバで実行される。これに対して、我々の Wiki Bot のプログラムは、Wiki Bot で実行される。

Obnize の場合も Kaseya や UNIFAS の場合と同様に、Cloud 側は特定の企業の製品に依存することになる。また、Obniz のデバイスはすべて Obniz Cloud を経由した通信によって制御されるため、Obnize Cloud に障害が発生すると利用できなくなる。これに対して、Wiki IoT は、複数の Wiki ソフトウェアと複数の Wiki ボットを使って、可用性を高めている。

5.4 NetNuclues Cloud Hub

東芝情報システム(株)の、NetNucleus Cloud Hub[19] も Obniz と同様に、WebSocket を使って、NAT の背後にある IoT デバイスをインターネット上のサーバから制御することを可能にしている。NetNuclues Cloud Hub の IoT デバイスは、C 言語を使って開発するため、Wiki IoT のように、Web ページ上のプログラムを、IoT デバイス側で実行させることも不可能ではないが、IoT デバイスの開発者側で、そのような機能を実装する必要がある。

6. おわりに

我々が従来から開発していた、Wiki ページをデータベースとして使う Wiki IoT を使って、外部からは直接アクセスできないサーバの生死を Twitter のつぶやきで確認したり、通常は出入りできないサーバ室の状況を、インターネット上の Web により監視したりするシステムを作成し、それらを利用していることについて述べた。IoT システムの利用場面において、センサ端末群から得られるデータを管理者が観察している最中に、データを取得する時間間隔を変更するなど、途中で設定を変えたい場合があるが、本 Wiki IoT は、センサ端末側に管理者が行かなくても、外部の Wiki の記述を変えるだけで設定変更が可能である。

本システムは NAT で守られた LAN 内で実行したり、Basic 認証が使えたりする以外は、なにもセキュリティ対

策を行っていない。センサが故障した場合の配慮もなされていない。今後これらの問題の解決を行っていく予定である。

謝辞 本研究の一部は JSPS 科研費 16K00197, 15H03055 の助成を受けて実施しました。

参考文献

- [1] 山之上 卓, 白澤竜馬, 小田謙太郎, 下園幸一. Wiki と携帯型遠隔操作端末を使った情報セキュリティ監視システム, 情報処理学会研究会報告, Vol. 2012-IOT-16, No.35,(2012).
- [2] Takashi Yamanoue, Kentaro Oda, Koichi Shimozono, A M2M system using Arduino, Android and Wiki Software, Proceedings of the 3rd IIAI International Conference on e-Services and Knowledge Management (IIAI ESKM 2012), pp.123-128, Fukuoka, Japan, 20-22 (Sep. 2012).
- [3] Takashi Yamanoue, Kentaro Oda, Koichi Shimozono. An Inter-Wiki Page Data Processor for a M2M System, 4th International Conference on E-Service and Knowledge Management (ESKM 2013), Advanced Applied Informatics (IIAIAAI), 2013 IIAI International Conference on., pp.45-50, Matsue, Japan., (2013).
- [4] PukiWiki, <https://en.wikipedia.org/wiki/PukiWiki>, as of Jan. 30, 2017.
- [5] 山之上卓, Twitter と Wiki を使った自動情報提示システム, 情報処理学会, 研究報告インターネットと運用技術 (IOT), vol. 2016-IOT-32, No.5, pp.1-7, (2016-03-03).
- [6] 山之上卓, 小田謙太郎, 下園幸一, Wiki で設定やプログラムを記述可能なセンサネットワークシステム, 情報処理学会研究報告, Vol.2013-IOT-23, No.2, (2013).
- [7] Raspberry Pi, <https://www.raspberrypi.org>, as of Jan. 30, 2017.
- [8] 山之上卓 "インターネット上の Wiki ページ上でエッジ側端末の I2C デバイスの設定と入出力を可能とする IoT システムの試作", インターネットと運用技術シンポジウム 2016 論文集, 情報処理学会シンポジウムシリーズ No. 2016, pp. 91-97, (2016-12)
- [9] 山之上卓, 羅 牧野, "センサネットワークのセンサ端末群をインターネット上の Wiki ページで制御する IoT システムの試作," (社)情報処理学会, 研究報告インターネットと運用技術 (IOT), vol. 2017-IOT-36, No. 12, pp. 1-8, Feb. 2017.
- [10] Android, <https://www.android.com> as of Jan. 30, 2017.
- [11] Arduino, <https://www.arduino.cc> as of Jan. 30, 2017.
- [12] New Relic <https://newrelic.com>
- [13] 情報倫理デジタルビデオ小品集 <https://axies.jp/ja/video>
- [14] Takashi Yamanoue, "Monitoring Servers, With a Little Help from my Bots", SIGUCCS '17 Proceedings of the 2017 ACM on SIGUCCS Annual Conference, pp. 173-180, Seattle, Washington, USA, Oct.,2017.
- [15] Panasonic, "焦電型赤外線センサ PaPIRs (パピルス)," <https://www3.panasonic.biz/ac/j/control/sensor/human/index.jsp>
- [16] KASEYA, <http://www.kaseya.com/>
- [17] UNIFAS, <http://www.furunsystems.co.jp/product/detail/unifas.html>
- [18] Obniz, <https://obniz.io>
- [19] NetNucleus Cloud Hub, https://www.tjsys.co.jp/embedded/netnucleus-cloudhub/index_j.htm