

# Wiki ページに書かれた R 言語のプログラム によるデータ解析を可能にした IoT システムの試作

平田 篤<sup>†1</sup>, 藤田 健吾<sup>†1</sup>, 伊勢本 和広<sup>†1</sup>, 山之上 卓<sup>†1</sup>

**概要:** Wiki ページに書かれた R 言語のプログラムによるデータ解析を可能にした IoT システムの試作について述べる. このシステムは今まで我々が開発していた Wiki と Bot を組み合わせた情報処理環境を R 言語のプログラムが使えるように拡張したものである. このシステムを使って授業中にどのくらいの学生が起きていて, どのくらいの学生が寝ているかをリアルタイムで教員に表示するシステムを開発中である.

**キーワード:** Bot, Wiki, R, IoT, 授業支援

## Experimental Implementation of an IoT system, Which Can Analysis Data by a Program in R language on a Wiki Page

ATSUSHI HIRATA<sup>†1</sup>, KENGO FUJITA<sup>†1</sup>, KAZUHIRO ISEMOTO<sup>†1</sup> and TAKASHI  
YAMANOUE<sup>†1</sup>

**Abstract:** This paper describes an experimental implementation of IoT system that enables data analysis using a R language program written in a Wiki page. This system is an extension of our previous information processing environment, which is a combination of Wiki and Bots, by adding R language program processor. We are developing a class assistant system which shows how many students are sleeping and how many students are getting up.

**Keywords:** Bot, Wiki, R, IoT, Class assisting

### 1. はじめに

我々は Wiki と Bot を組み合わせた情報処理環境の開発を行っている[2][4]. 今までにこの情報処理環境を使って Wiki に書かれたコマンドやプログラムによって Bot を制御し, 他の Wiki ページに書かれたデータを読み込んで計算を行い, 結果を Wiki ページに書き込むシステムなどを開発した. また, この環境の応用として LAN の情報セキュリティを強化するためのシステムも開発している[1][3].

今回, 今までこの環境が備えていたプログラミング言語に加えて, 統計解析向けプログラミング言語 R のプログラムをこの環境で実行するための Bot を試作した. IoT システムは多くのセンサによって得られたデータを解析し有益な情報を人間に示すものでもある. 大量のデータを解析し, 有益な情報を得るために R を使ったデータ解析が行われてきた. そのためデータ解析のための大量の R プログラムが存在し, 多くの人が R プログラムを記述することが出来る. 今回試作を行った Bot により本情報処理環境でこれらの大量の R プログラムが利用できるようになり, 多くの人が本情報処理環境の上で R を使った IoT システムのデータ解析を行えるようになる. これらの R プログラムは Wiki ペー

ジの上で記述されるので, 特定または不特定のグループ内でこれらの R プログラムを共有したり, R を使った IoT システム全体を共同開発したりすることが出来る.

また, Raspberry Pi に PIR センサを接続した Bot も開発した. この Bot を使って, 近くの人々の活動状況を定期的に Wiki のページに書き込むことが可能になる.

現在, R 言語プログラムの処理機能を持った Bot と PIR センサを接続した Raspberry Pi の Bot を組み合わせて, 授業中にどのくらいの学生が起きていて, どのくらいの学生が寝ているかをリアルタイムで教員に表示するシステム [6]を開発中である.

### 2. システムの概要

現在開発中の, 授業中の学生の活動状況を教員に実時間で表示するシステムの概要の例を図 1 に示す. このシステムは, Bot②・③・④が, それらについている PIR センサを使って, 一定時間ごとに学生の活動状況のデータを収集して, PukiWiki②・③・④のページに書き込み, Bot①がそのデータを解析して, PukiWiki①にその結果を書き込み, 教師に表示する.

<sup>†1</sup> 福山大学  
Fukuyama University

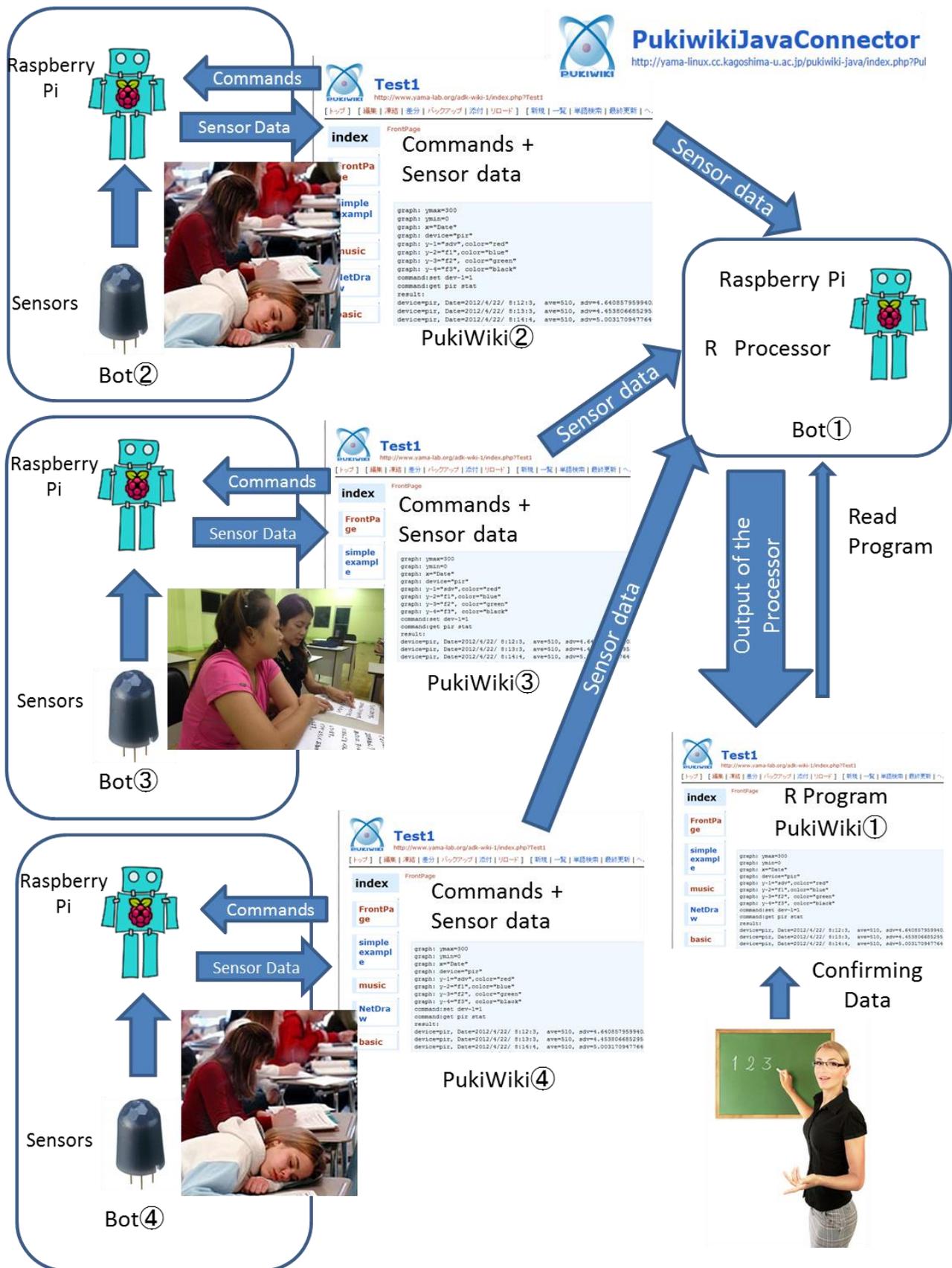


図1 システムの概要

Bot①は一定時間ごとに PukiWiki①のページ上のコマンドやプログラムを読み込んで実行する。このページには、PukiWiki②・③・④のページに書かれたデータを読み込む Basic プログラムと、読み込まれたデータの統計計算を行

い、結果を PukiWiki①のページに書き込む R 言語プログラムが記述されている。Bot②・③・④は、それぞれが持っている PIR センサで収集したデータを PukiWiki②・③・④のページに定期的な書き込む。

## test4

<http://www.yama-lab.org/r-1/index.php?test4>

[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](#) |

### FrontPage

```
command: R ex1
R: print("start")
command: end ex1
command: program ex2
program: dim urls
program: dim Table
program: dim RowLabel
program: dim ColumnLabel
program: n=3
program: vals="c("
program: urls(0)="http://www.yama-lab.org/r-1/index.php?test3"
program: urls(1)="http://www.yama-lab.org/r-1/index.php?test5"
program: urls(2)="http://www.yama-lab.org/r-1/index.php?test6"
program: for i=0 to n-2
program:   page=ex("connector", "getpage "+urls(i))
program:   pageR=getResultPart(page)
program:   parseCsv(pageR, Table, RowLabel, ColumnLabel)
program:   ix=getindex(Table, RowLabel, ColumnLabel("device"), "=", "a-0")
program:   vals=vals+Table(ix, ColumnLabel("v"))+", "
program: next i
program: page=ex("connector", "getpage "+urls(n-1))
program: pageR=getResultPart(page)
program: parseCsv(pageR, Table, RowLabel, ColumnLabel)
program: ix=getindex(Table, RowLabel, ColumnLabel("device"), "=", "a-0")
program: vals=vals+Table(ix, ColumnLabel("v"))+" "
program: ex("ex1", "run")
program: out="eval v <- "+vals
program: ex("ex1", out)
program: ex("ex1", "eval mean(v)")
command: end ex2
command: run ex2
command: clear sendBuffer
command: getOutput ex1
command: sendResults.
result:
[STRING "start"]
[REAL* (0.0, 3.0, 2.0)]
[REAL* (0.0, 3.0, 2.0)]
[REAL* (1.6666666666666667)]
currentDevice="yamaRasPiDp9_1", Date=2016/9/6/ 13:1:30
```

図2 R 言語のプログラムとその実行結果が書かれた Wiki ページ

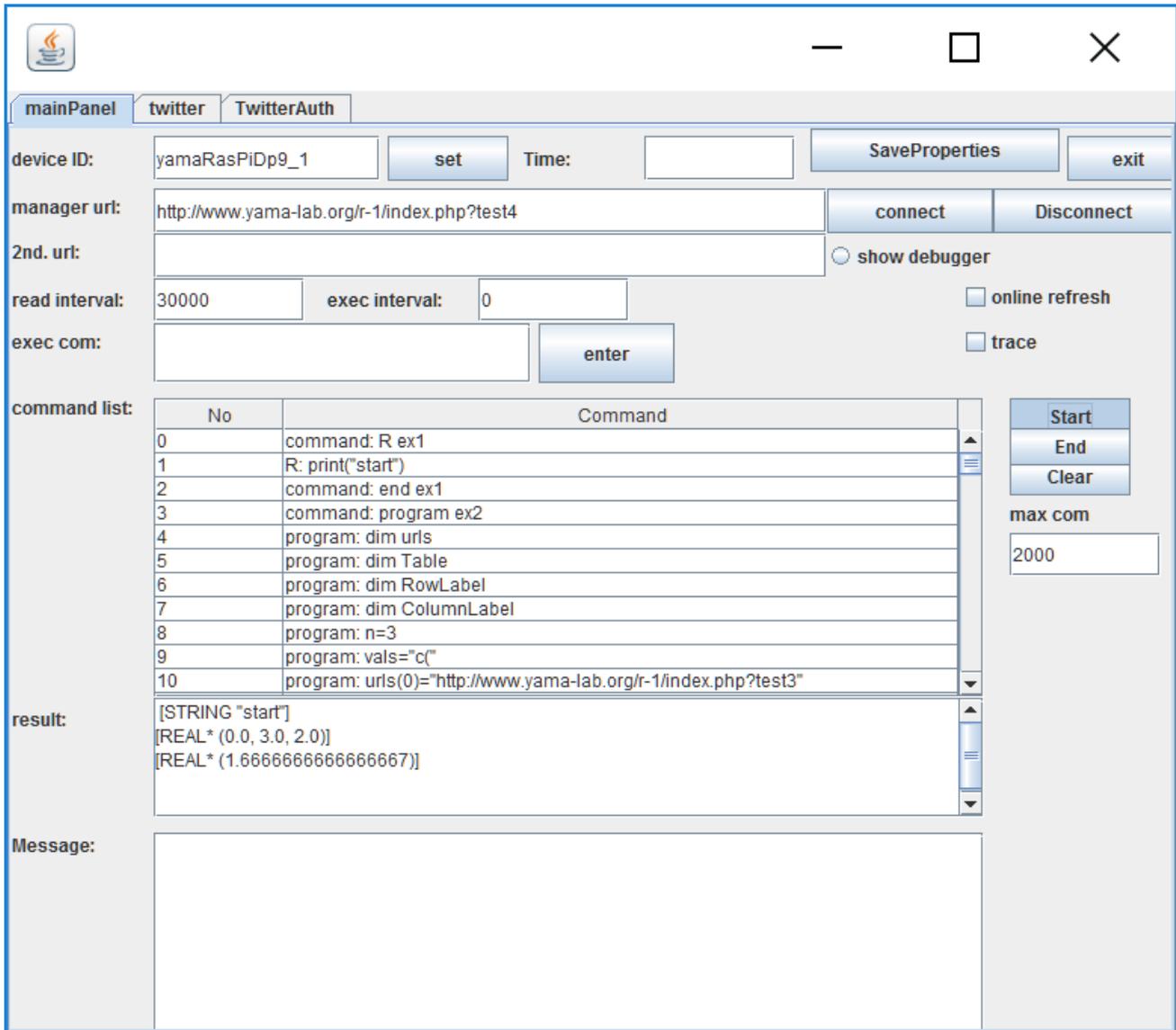


図3 BotによるWikiの読み込み

### 3. Rプログラムの実行例

ここでは図1の、授業中の学生の活動状況を教員に実時間で表示するシステムの、PukiWiki①とBot①の実行例について示す。

このWikiページのプログラムの入力ブラウザで該当のWikiページを開き、Wikiが持っているエディターを使って入力する。入力されたRのプログラムとBasicのプログラムとコマンド列と出力の例を図2に示す。

Botが定期的にこのページを読み込み、Rのプログラムとコマンド列を実行する。このBotが実行を終えると図2のWikiページのresult:の行以降に実行結果が書き込まれる。図3に図2に書かれたプログラムを実行するBotのGUIを示す。

図2のresult:の行より下にRプログラムの実行結果が書き込まれている。この中で

[REAL\* (1.6666666666666667)]

の行が教師に示される授業中の学生の活動状況でありこれは全学生の活動状況の平均値を示している。

また、Rの変数に値が代入される度に結果が表示されるため[REAL\*(0.0,3.0,2.0)]の行が2度表示されている。

以下に図2の各行の説明を記す

- command: R ex1  
この行以降にRのプログラムが記述されることを示す。このRプログラムの名前をex1とする。
- R: print("start")  
Startを表示するRのプログラム
- command: end ex1  
プログラムex1がここで終わることを示す
- command: program ex2  
この行以降にBasicのプログラムが記述されることを示す

示す. この Basic のプログラムの名前を ex2 とする.

- program: dim urls  
url を入れる配列を作る.
  - program: dim Table  
他の Wiki ページに書かれた CSV 形式のデータを格納する表を表す配列
  - program: dim RowLabel  
上の表の列の情報を示す配列.
  - program: dim ColumnLabel  
上の表の行の情報を示す配列.
  - program: n=3  
n は他の Wiki ページの数
  - program: vals="c"  
R の実行環境に値を引き渡す準備
  - program: urls(0)=<http://www.yama-lab.org/r-1/index.php?test3>  
1 番目の Wiki ページの url を配列 urls の 0 番目に入れる. 2 番目, 3 番目の Wiki ページについても同様
  - program: for i=0 to n-2  
この行の下の行から next i の上の行までを i を 0 から n-2 まで変化させて繰り返し実行する.
  - program: page=ex("connector", "getpage "+urls(i))  
urls(i)に格納された Wiki ページを読み込んで page に入れる
  - program: pageR=getResultPart(page)  
上で読み込んだ Wiki ページの Result:の行以降を pageR に入れる
  - program: parseCsv(pageR, Table, RowLabel, ColumnLabel)  
この行は文字列 pageR を二次元配列 Table に読み込む. pageR は以下のように等式をカンマで区切って並べた行を並べたものであることを仮定している.  

$$[col-label_1]=[val_{1-1}], \dots, [col-label_{1-n}]=[val_{1-n}].$$

$$[col-label_2]=[val_{2-1}], \dots, [col-label_{2-2n}]=[val_{2-2n}].$$

...

$$[col-label_m]=[val_{1-m}], \dots, [col-label_{m-mn}]=[val_{m-mn}].$$
- 各行は同じ数の等式を持たなくても良い. 以下に例を示す.
- ```
device=d, Date=2013/5/5/ 17:6:18, v=0x0c0.
device=a-2, Date=2013/5/5/ 17:6:18, v=155.
device=a-1, Date=2013/5/5/ 17:6:18, v=53.
device=a-0, Date=2013/5/5/ 17:6:45, ave=242, ...,
dt=100.
device=a-0, Date=2013/5/5/ 17:7:53, ave=242, ...,
dt=100.
```

...

- RowLabel は("rowcol", "row") と ("maxIndex", 表の行数)の見出しと内容のペアを格納したハッシュ表である. ColumnLabel は("rowcol","col"), ("maxIndex", 表の列数), ([col-label-1], 表におけるこの Label の列の位置), ..., ([col-label-max], 表におけるこの Label の列の位置)の見出しと情報のペアを格納したハッシュ表である.
- program: ix=getindex( Table, RowLabel, ColumnLabel("device"), "=", "a-0")  
列の Label が"device"でその値が"a-0"になるような最初の行の位置を ix に入れる.
- program: vals=vals+Table(ix,ColumnLabel("v"))+,"  
ix の行の Label が"v"である列の値を取り出し, 文字列 vals の後ろに文字列として連結し, ","をその後ろに連結する.
- program: page=ex("connector", "getpage "+urls(n-1))  
program: pageR=getResultPart(page)  
program: parseCsv(pageR, Table, RowLabel, ColumnLabel)  
program: ix=getindex(Table, RowLabel, ColumnLabel("device"), "=", "a-0")  
program: vals=vals+Table(ix,ColumnLabel("v"))+,"  
最後に読み込むページについて上と同様のことをやり, vals の後ろに")"を連結する
- program: ex("ex1", "run")  
プログラム ex1 を実行する.
- program: out="eval v <- "+vals  
program: ex("ex1", out)  
R プログラム ex1 の実行環境で  
v <- c(v1,v2,v3)  
を実行する. ここで v1,v2,v3 は他の 3 つの Wiki ページに書かれたデータである.
- program: ex("ex1", "eval mean(v)")  
R プログラム ex1 の実行環境で v の値の平均を出す.
- command: end ex2  
Basic プログラム ex2 の終わりを表す.
- command: run ex2  
プログラム ex2 を実行する.
- command: clear sendBuffer  
出力 Buffer の内容を消去する.
- command: getOutput ex1  
R のプログラムの実行結果を出力 Buffer に追加する.
- command: sendResults.  
出力 Buffer の内容を Wiki ページの result:以下に書き込む.

#### 4. Raspberry Pi とセンサで構成した Bot

図 4 に、図 1 における Bot②・③・④の例を示す。この Bot は Raspberry Pi に、PIR センサ(Passive Infrared センサ)を接続して構成されている。Raspberry Pi はネットワークに接続されており、PIR センサは Raspberry Pi の GPIO に接続されている。

この Bot は、起動時に設定された PukiWiki②・③・④のページを定期的に読み込む。そこに書かれたコマンドやプログラムによって、この Bot の動きが制御される。

PukiWiki のページを読み込む間隔とは独立に、この Bot は一定間隔ごとに PIR のデジタル出力（人の動きがなければ 0、あれば 1）を読み込んで、一定期間ごとに何度 1 があつたか計測している。この値は PIR の近くに人がいた場合の、人の活動状況に関する一つの指標の値となる。この結果、一定時間ごとに、センサの近くの人々の活動状況が、PukiWiki のページに書き込まれることになる。図 5 に、Bot に対して、人の活動状況を収集することを指示したコマンドと、実際に収集されたデータが記載されたページの例を示す。以下に図 5 の各行の説明を示す。

- objectPage <http://192.168.2.107/pukiwiki/index.php?Test02>  
自分のページの URL.
- device yamaRasPiDp1\_1 or yamaRasPiDp1\_2 start after no write for 10 min.  
Bot の一台が何らかの理由で停止した時にもう一つの Bot を動かす。
- command: set readInterval=10000  
このページを Bot が読む間隔。
- command: clear sendBuffer  
result:以降の行を一回削除する。
- command: get in-a-0

PIR から人の動きの状況を入力して、日付とともに result:以下に書き込む。

- command: set sendInterval=10000  
result:以降に出力結果を書き込む間隔。
- result:  
この行以降に出力結果が書き込まれる。
- device=a-0, Date=2016/9/14/ 14:51:46, v=1.  
書き込まれた出力結果。
- currentDevice="",Date=2016/9/14/ 14:51:47  
Bot の名前と Bot が最後に書き込んだ日時。  
ここでは Bot に名前がついていない。

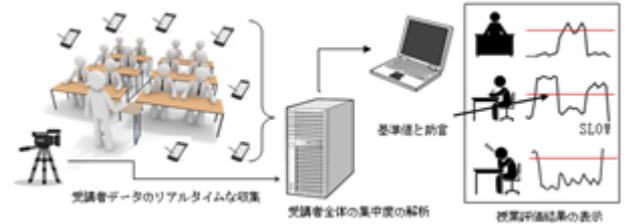


図 4 教室内の学生の居眠り状況を教師が実時間で把握するためのシステム[6]



図 5 Raspberry Pi に、センサとして PIR センサを接続して構成した Bot

## Test02

<http://192.168.2.107/pukiwiki/index.php?Test02>

編集 | 凍結 | 差分 | バックアップ | 添付 | リロード ] [ 新規 | 一覧 | 単

```

objectPage http://192.168.2.107/pukiwiki/index.php?Test02
device yamaRasPiDp1_1 or yamaRasPiDp1_2 start after no write for 10 min.
command: set readInterval=10000
command: clear sendBuffer
command: get in-a-0
command: set sendInterval=10000
command: set execInterval=10000
result:
device=a-0, Date=2016/8/29/ 17:42:59, v=1.
currentDevice="", Date=2016/8/29/ 17:43:3

```

図 6 Bot に対して人の活動状況を収集することを指示したコマンドと、収集されたデータが記載された PukiWiki のページ

## 5. 関連研究

Web を使ってパソコンを管理するための遠隔操作システムとして Kaseya[5]や UNIFAS[7]がある。Kaseya は、多くの PC やサーバーの遠隔保守管理を行うものであり、UNIFAS は多くの無線 LAN アクセスポイントの遠隔保守管理を行うものである。これに対して我々の環境では Web を使って Bot を遠隔操作するものであるが、Web で遠隔操作をする面では類似している。なお、Kaseya や UNIFAS では PukiWiki のページに記載されている R 言語のプログラムを PC や無線 LAN アクセスポイントで実行させることは出来ない。

IEEE1888[8]は、次世代 BEMS やスマートグリッド向けに開発され、2011 年に国際標準化されたオープンな通信規格であり、IoT システムに関する標準的な通信プロトコルの 1 つである。IEEE1888 は IoT デバイスとインターネット上のストレージ間の通信を規定するものであり、IoT デバイスの遠隔操作を行うことについての規定はない。

## 6. おわりに

Wiki ページに書かれた統計解析向けプログラミング言語 R のプログラムを実行可能な Bot と、Raspberry Pi に PIR センサを接続した Bot の開発と、この 2 つの Bot を使って、授業中にどのくらいの学生が起きていて、どのくらいの学生が寝ているかをリアルタイムで教員に表示する授業支援システムについて述べた。現在、このシステムについて、少数のセンサ側 Bot と R 側 Bot を繋いで動かすことが出来た。今後、この授業支援システムの試作を進めるとともに、実際の授業への応用を行う予定である。

また、このシステムを 15 個ほど用意し実際の授業で動かしてみる予定である。

現在、このシステムはセキュリティについてはなにも考慮していない。今後セキュリティについても考慮する必要がある。

## 謝辞

本研究の一部は JSPS 科研費 JP16K00197 の助成を受けた。

## 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. A Malicious Bot Capturing System using a Beneficial Bot and Wiki, Journal of Information Processing(JIP), vol.21, No.2, pp.237-245(2013).
- [4] 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).
- [5] Kaseya <http://kaseya.blue.co.jp/>
  - [6] 尾関孝史, 渡邊栄治, 電子ツールを用いた講師の板書と受講者のノートに対する時系列解析, 信学技報 ET2016-31, pp.55-60, 2016.
  - [7] UNIFAS <http://www.furunsystems.co.jp/product/detail/unifas.html>
  - [8] IEEE1888 <http://www.gutp.jp/fiap/>